

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

О.В. Коваль

(підпис)

(ініціали, прізвище)

“ ” 2019р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 “Комп’ютерні науки”

на тему: «Моделювання процесу обтікання крилового профілю на малих швидкостях»

Виконала: студентка IV курсу, групи ТР-51

Онісімчук Мар’яна Вікторівна

(прізвище, ім’я, по батькові)

(підпис)

Керівник проф. д.ф.-м.н., Гуржій О.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Консультант

(назва розділу)

(вчені ступінь та звання, прізвище, ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ
Завідувач кафедри
О.В. Коваль
(підпис)
” __ ” _____ 2019р.

ЗАВДАННЯ
на дипломну роботу студенту
Онiсiмчук Мар’яні Вікторівні
(прізвище, ім’я, по батькові)

1. Тема роботи: «Моделювання процесу обтікання крилового профілю на малих швидкостях»

керівник роботи проф. д.ф.-м.н. Гуржій Олександр Андрійович
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” __ ” __ 201__ р. № __

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування Python, середовище PyCharm Community Edition 2019, система керування базами даних MongoDB

4. Об’єкт дослідження комп’ютерні технології моделювання крилових профілів

5. Перелік питань, які потрібно розробити: проаналізувати існуючі методи моделювання крилових профілів, розробити алгоритми для моделювання аеродинамічних характеристик на малих швидкостях польоту, розробити програмний комплекс, який дозволяє для заданої геометрії поперечного перерізу крила розрахувати аеродинамічні характеристики

6. Перелік ілюстративного матеріалу розподіл поля швидкостей та тиску навколо крилового профілю, схема розташування точкових вихрів та точок колокації, структура програмного комплексу, архітектура програмного комплексу, приклади виклику підсистем програми

7. Дата видачі завдання "10" жовтня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	09.10.18	
2.	Вивчення та аналіз задачі	15.10. – 30.11.18	
3.	Дослідження процесу моделювання крилових профілів	1.12.18 – 27.01.19	
4.	Аналіз існуючих програмних рішень	28.01 – 24.02.18	
5.	Розробка архітектури та структури програми	25.02 – 17.03.18	
6.	Проектування архітектури програмної системи	18.03 – 17.04.18	
7.	Програмна реалізація системи моделювання процесу обтікання крилових профілів	18.04 – 19.05.18	
8.	Оформлення пояснювальної записки	02.05 – 16.06.19	
9.	Захист програмного продукту	20.05.19	
10.	Передзахист	28.05.19	
11.	Захист	19.06.19	

Студент

(підпис)

Керівник роботи

(підпис)

Онісімчук М.В.

(прізвище та ініціали,)

Гуржій О. А.

(прізвище та ініціали,)

АНОТАЦІЯ

Метою дипломної роботи було створення програмної системи спрямованої на моделювання обтікання крилового профілю з різним кутом нахилу та обчислення аеродинамічних характеристик для оцінки якості крила на малих швидкостях. Система генерує обтікання будь якого профілю відповідно завантажених координат та заданих параметрів обчислень. Створено керуючий модуль системи, який реалізований за допомогою основних законів аеродинаміки і методу дискретних особливостей. Користувацький додаток представляє собою демонстрацію роботи розробленого алгоритму.

Записка містить 80 сторінок 21 рисунок 49 формул та 25 посилань.

Ключові слова: Python, моделювання, аеродинаміка, крило, якість крила, обтікання, функція току.

ABSTRACT

The purpose of the thesis was to create a software system designed to simulate profile wrap with different angle of inclination and aerodynamic performance calculations to assess the quality of the wing at low speeds. The system generates the flow of any profile of the loaded coordinates and the specified computing parameters. A control module of the system is created, which is realized with the help of the basic laws of aerodynamics and method of discrete features. The custom application is a demonstration of the work of the developed algorithm.

The note contains 80 pages 21 images 49 formulas and 25 references.

Keywords: Python, modeling, aerodynamics, wing, wing quality, flow, current function.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1. ПОСТАНОВКА ЗАДАЧІ	10
2. ДОСЛІДЖЕННЯ ПРОЦЕСУ АЕРОДИНАМІЧНОГО ОБТІКАННЯ КРИЛОВОГО ПРОФІЛЮ	11
2.1 Проблематика досліджень в аеродинаміці	11
2.2 Засоби досліджень в аеродинаміці	12
2.2.1 Теоретична складова аеродинаміки	12
2.2.2 Комп'ютерна аеродинаміка	17
2.3 Дослідження аеродинаміки крила	19
2.3.1 Режими обтікання	20
2.3.2 Дозвукове обтікання крила	20
2.3.3 Виникнення підйімальної сили.	23
2.3.4 Виникнення лобового опору. Формула лобового опору	27
2.3.5 Аеродинамічна якість крила	29
2.4 Висновки до другого розділу	30
3. МОДЕЛЮВАННЯ ПРОЦЕСУ АЕРОДИНАМІЧНОГО ОБТІКАННЯ КРИЛА	31
3.1 Аналіз методів для вирішення задачі знаходження аеродинамічних характеристик	31
3.1.1 Метод скінчених елементів	34
3.1.2 Метод граничних елементів	35
3.1.3 Метод Гальоркіна	36
3.1.4 Метод Рітца	38
3.1.5 Метод дискретних особливостей	38
3.2 Використання методів обчислень для розв'язку математичної задачі процесу моделювання обтікання крилового профілю	41
3.2.1 Принцип розміщення точок колокації	42
3.2.2 Метод Гауса для СЛАР	43
3.2.3 Метод Рунге-Кутта-Фельберга 4-го порядку	46
3.3. Висновки до третього розділу	48

4. ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНОЇ СИСТЕМИ	49
4.1 Мова програмування Python	49
4.1.1 Об'єктно-орієнтоване програмування	50
4.1.2 Функціональне програмування	51
4.1.3 Модулі та пакети	51
4.1.4 Інтроспекція	52
4.1.5 Ітератори	52
4.1.6 Генератори	52
4.1.7 Керування контекстом виконання	53
4.1.8 Декоратори	53
4.1.9 Бібліотеки	54
4.2 Математичні обчислення з використанням бібліотеки Numpy	57
4.3 Візуалізація результатів за допомогою бібліотеки matplotlib	58
4.4 Розробка підсистеми збереження інформації	59
4.4.1 Збереження інформації до бази даних MongoDB	60
4.4.2 Формат даних JSON	61
4.5 Висновки до четвертого розділу	63
5. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ	64
5.1 Функціональна модель	64
5.2 Внутрішнє представлення даних	70
5.3 Структура програми	70
5.4 Висновки до п'ятого розділу	73
6. ПРИНЦИП РОБОТИ З ПРОГРАМНОЮ СИСТЕМОЮ	74
6.1 Системні вимоги	74
6.2. Порядок роботи користувача з системою	74
6.3 Висновки до шостого розділу	77
ВИСНОВКИ	78
СПИСОК ДЖЕРЕЛ	79
ДОДАТОК А	82
ДОДАТОК Б	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЧДР — часткове диференціальне рівняння;

БД — база даних;

СКБД — система керування базами даних;

API (англ. Application Programming Interface) — прикладний програмний інтерфейс;

ВСТУП

З кожним роком все більше виникає необхідність у використанні малої авіації. Мала авіаційна техніка володіє мобільністю, високим рівнем безпеки, зручна в управлінні та використанні, комфортна для пасажирів і має ряд інших переваг. Сьогодні послугами малої авіації широко використовують у сільському господарстві для використання ефективної хімічної, біологічної обробки угідь і потрібні під час здійснення різних видів рятувальних робіт. Її подальший розвиток дає змогу розглядати як альтернативу наземному транспорту. Легкі літаки не тільки перевозять пасажирів і вантажі, але також беруть участь в патрулюванні лісів, нафтогазопроводів і енергомереж. На них навчаються пілоти та використовують в поліції та авіаспорті.

В останні роки інтенсивний розвиток малої авіації ставить перед розробниками широкий перелік наукових і конструкторських проблем, одна з яких пов'язана з розробкою моделі крилових профілів для оцінки та удосконалення їх характеристик на малих швидкостях польоту. Аналіз світової наукової літератури показує, що розробці ефективних крилових профілів для малих швидкісних режимів сьогодні приділяється мало уваги.

1. ПОСТАНОВКА ЗАДАЧІ

Метою дипломної роботи є розробка програмної системи, спрямованої на моделювання процесу обтікання крилового профілю на малих швидкостях та знаходження аеродинамічних характеристик крилового профілю для оцінки якості крила, виявлення закономірностей щодо моделювання аеродинамічних процесів.

Аналіз математичних методів повинен забезпечити пошук найоптимальнішого рішення для розв'язку поставленої задачі.

Об'єктом дослідження є комп'ютерні технології проектування крилових профілів.

Предмет дослідження – комп'ютерні технології моделювання процесу обтікання крилового профілю та аеродинаміки крилових профілів на малих швидкостях.

Розроблена система повинна забезпечувати наступні можливості:

- побудова траєкторії обтікання профілю крила, побудованого за заданими координатами;
- реалізація зміни кута атаки та швидкості потоку газу;
- візуалізацію отриманих результатів;
- розрахунок оцінки якості крила;

2. ДОСЛІДЖЕННЯ ПРОЦЕСУ АЕРОДИНАМІЧНОГО ОБТІКАННЯ КРИЛОВОГО ПРОФІЛЮ

2.1 Проблематика досліджень в аеродинаміці

Дослідження в аеродинаміці пов'язані з визначенням наступних характеристик потоку: густина, швидкість частинок середовища, температура як функція простору та часу, тиск. В конкретній ситуації можна обчислити силу та момент сили, що діє на тіло у потоці після того як визначимо всі вище перелічені характеристики. Аеродинаміка як наукова дисципліна служить базою для вирішення широкого спектру прикладних задач.

Існує перелік проблем на практиці, які вирішуються за допомогою конкретних задач аеродинаміки. Він включає не тільки проблеми авіації, а й проблеми морського та наземного транспорту, проблеми ракетобудування, екології, метеорології, сільського господарства та інші. Щоб отримати відповіді на запитання в аеродинаміці використовують теоретичні та експериментальні методи. Сьогодні широко застосовуються методи комп'ютерної аеродинаміки. У дослідників є достатньо потужні засоби, щоб в рамках математичної моделі одержати кількісні оцінки характеристик течій. Проте методами комп'ютерної динаміки рідин та газів здобувається великий об'єм інформації, який для розуміння та аналізу вимагає від дослідника повного розуміння фізики процесів та причинно-наслідкових зв'язків. Саме ця причина уможливила появу досить детальних досліджень змісту понять, які, як ми думали, є давно установленими.

2.2 Засоби досліджень в аеродинаміці

Для встановлення загальних висновків про характер течії, основуючись на певних припущеннях формують відповідні математичні моделі.

При розв'язанні різних математичних задач можуть виникати труднощі і дані про конкретні випадки обтікання тіл одержують з допомогою експерименту, часто досить складного і дорогого. В результаті розвитку комп'ютерної техніки відкрились принципово нові можливості одержання кількісних оцінок характеристик потоків. На основі цього сформовано новий розділ механіки – комп'ютерна динаміка газу та рідин.

2.2.1 Теоретична складова аеродинаміки

На сьогоднішній день в аеродинаміці розроблено багато математичних моделей, які призначені для дослідження потоків в залежності від їх властивостей та швидкості. Дані моделі сформовані на базі загальних законів збереження, але використання тільки цих законів не дає змогу сформулювати систему рівнянь для опису потоку у повному вигляді. Окрім законів збереження потрібно застосовувати рівняння стану, що описують зв'язок між силовими, кінематичними характеристиками і температурою в газі для конкретних умов існування потоку.

Модель нестисливого ідеального газу є найпростішою для використання в аеродинаміці. При її застосуванні у газі відсутня в'язкість, зсувні напруження, теплопровідність. Виходячи з цих умов можна зрозуміти головні моменти процесу побудови математичної моделі в механіці суцільного середовища., Використовуючи Декартову систему координат розглянемо основні рівняння, припускаючи розгляд ідеального газу (рідини). Окрім рівняння нерозривності при дослідженні руху частинок середовища доцільно використовувати рівняння другого закону Ньютона. В ідеальному газі відсутній опір при зсувному русі одного шару газу відносно іншого, вектор зусиль на будь-якій поверхні всередині середовища перпендикулярний до

поверхні і має в даній точці постійну величину при зміні орієнтації площини. По суті це означає ігнорування в'язкості газу, яка притаманна кожній реальній рідині чи газу.

Перш за все слід взяти до уваги найважливіше співвідношення, яке доповнює загальні закони збереження та присутнє в усіх моделях механіки суцільного середовища — це **рівняння нерозривності**. Його можна розглядати як конкретну форму запису закону збереження маси. Для складових вектору швидкості частинок середовища застосовуємо Декартові компоненти U, V, W , а густину середовища позначимо як $\rho = \rho(x, y, z, t)$, тоді рівняння нерозривності матиме вигляд:

$$\frac{\partial (\rho U)}{\partial x} + \frac{\partial (\rho V)}{\partial y} + \frac{\partial (\rho W)}{\partial z} = - \frac{\partial \rho}{\partial t}. \quad (1.1)$$

Звідси, притік рідини або газу у виділений об'єм покриває заміну густини середовища. Це співвідношення записане в змінних Ейлера, оскільки формується при розгляді сталого об'єму.

Вигляд рівняння нерозривності незмінний для всіх середовищ – металу, води, повітря. Воно має відносно універсальний характер, тому повинно прийматися до уваги при математичному моделюванні будь-яких суцільних середовищ. Якщо при русі середовища потрібно враховувати його стисливість, дане рівняння включає чотири характеристики стану середовища — три компоненти вектора швидкості та густину.

Рівняння Ейлера. Розглянемо відношення другого закону Ньютона для елементарного об'єму газу в вигляді паралелепіпеда, сторони якого dx, dy, dz , в формі:

$$\rho \frac{dV}{dt} = \rho F - \text{grad} P$$

де F – об'ємна сила і використовується загально прийняте позначення для оператора градієнта скалярної величини тиску P .

Оскільки рівняння записано відносно конкретної матеріальної частинки, то при диференціюванні вектора швидкості слід враховувати залежність від часу координат частинки речовини в векторі швидкості $V = V(x(t), y(t), z(t), t)$

Отже, в координатній формі три рівняння руху частинки середовища записуються так:

$$\begin{aligned}\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} + W \frac{\partial U}{\partial z} &= F_x - \frac{1}{\rho} \frac{\partial P}{\partial x}, \\ \frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + W \frac{\partial V}{\partial z} &= F_y - \frac{1}{\rho} \frac{\partial P}{\partial y}, \\ \frac{\partial W}{\partial t} + U \frac{\partial W}{\partial x} + V \frac{\partial W}{\partial y} + W \frac{\partial W}{\partial z} &= F_z - \frac{1}{\rho} \frac{\partial P}{\partial z}.\end{aligned}\tag{1.2}$$

Ці рівняння називаються **рівняннями Ейлера**. Система рівнянь (1.1) і (1.2) має чотири рівняння відносно п'яти невідомих функцій, показує її незамкненість (неповноту) при відповідних об'ємних силах. Фізичні властивості середовища визначаються додатковим рівнянням, яке доповнюють систему. Через це вони називаються рівняннями стану. Досить часто рівняння стану описують складні процеси, що відбуваються в потоці. Прості види таких рівнянь використовуються для розуміння певних найпростіших явищ, пов'язаних з рухом газу. Таким рівнянням може бути співвідношення, яке показує припущення про те, що в процесі руху рідини її густина залишається незмінною величиною. Математично це записується рівнянням:

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + U \frac{\partial \rho}{\partial x} + V \frac{\partial \rho}{\partial y} + W \frac{\partial \rho}{\partial z}.\tag{1.3}$$

Тепер система рівнянь (1.1) – (1.3) замкнена. Вона і формує основні співвідношення математичної моделі ідеального нестисливої рідини. Для одержання кількісних характеристик потоку ця система рівнянь має бути розв'язаною при певних початкових та граничних умовах. Математичні моделі для різних видів течій в аеродинаміці будуються саме за такою технологією. Потрібно брати до уваги, що побудова математичної моделі процесу, який досліджується, є головним етапом в

вирішенні задач прикладної та теоретичної аеродинаміки. Оскільки рівнянь руху (1.2) є нелінійним, виникають труднощі при одержанні розв'язків відповідних математичних задач. На сьогодні існує значна кількість повних математичних моделей для опису аеродинамічних процесів.

Дослідження потоку на малих швидкостях в основному ґрунтуються на **рівнянні Нав'є-Стокса**. Це система диференціальних рівнянь у часткових похідних, яке описує теплопередачу і рух в'язкої н'ютонової рідини. Рівняння Нав'є-Стокса є найважливішим у гідродинаміці та застосовується у математичних моделях різних природних явищ.

Система містить два рівняння:

- рівняння нерозривності,
- рівняння руху.

Записуються наступним чином у векторному вигляді:

$$\frac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla) \vec{v} + \nu \Delta \vec{v} - \frac{1}{\rho} \nabla P + \vec{f}, \quad (1.4)$$

$$\nabla \cdot \vec{v} = 0,$$

де Δ — оператор Лапласа, ∇ — оператор Гамільтона, ν — коефіцієнт кінематичної в'язкості, t — час, ρ — щільність, $\vec{v} = (v^1, \dots, v^n)$ — векторне поле швидкостей, P — тиск, \vec{f} — векторне поле масових сил. P і v — функції часу t які нам невідомі і координати $x \in \Omega$, де $\Omega \subset R^n, n = 2, 3$ — тривимірна або плоска область, в якій рухається рідина. В систему даних рівнянь прийнято додавати початкові й крайові умови, наприклад:

$$\vec{v}|_{\partial\Omega} = 0,$$

$$\vec{v}|_{t=0} = \vec{v}_0.$$

Суть однієї з відкритих проблем полягає в аналізі знаходження розв'язків рівнянь. Задача знаходження загального розв'язання системи ускладнюється тим, що воно сильно залежить від початкових і граничних умов і є нелінійним.

Закон Бернуллі. Для проведення якісного аналізу особливостей потоків важливу роль має запис закону збереження енергії у величинах, що входять в загальні рівняння аеродинаміки. Таке співвідношення називається *законом Бернуллі*, та має вигляд:

$$\rho g h + P + \frac{\rho |V|^2}{2} = \text{const} \quad (1.5)$$

Прийmemo до уваги те, що розглядається ідеальний газ. В такому випадку робота внутрішніх сил в газі дорівнює нулю. Вважаємо, що між стінками крила і газом відсутнє тертя.

Загальний тиск в потоці газу – це сума динамічного і статичного тисків. Статичний тиск $(P + \rho g h)$ визначений потенційною енергією газу, що знаходиться під дією тиску і представляє собою суму тиску, обумовленого зовнішніми силами, та тиску, що обумовлений вагою об'єму виділеного газу. Динамічний $(\rho U^2/2)$ тиск визначається кінетичною енергією газу.

Згідно рівняння Бернуллі, при стаціонарній течії ідеального газу сума динамічного і статичного тисків стала в будь якій точці обтікання крилового профілю. Подальше використання даного рівняння дозволяє здійснити оцінку розподілу тиску на крило. Оскільки для крила висота незмінна, рівняння Бернуллі набуває наступного вигляду:

$$P_0 + \rho \frac{U_0^2}{2} = P(x, y) + \rho \frac{U^2(x, y)}{2}, \quad (1.6)$$

Де P_0 – атмосферний тиск, ρ – густина повітря, P – тиск в поточній точці, U – швидкість потоку в поточній точці, U_0 – початкова швидкість течії повітря.

2.2.2 Комп'ютерна аеродинаміка

Комп'ютерна аеродинаміка – один з розділів аеродинаміки, математичні методи якого засновані на підходах теоретичної аеродинаміки. При цьому методи розв'язання початково-граничних задач достатньо своєрідні. Розділ дискретної математики практично сформований на відмові від застосування методів аналізу, які базуються на асимптотичних відношеннях нескінченно малих величин. Методів дискретного аналізу відкриває можливості для отримання характеристики потоків, характеристики яких неможливо отримати в рамках існуючих аналітичних рішень. Широке застосування даних методів в механіці суцільного середовища стало причиною формування нового напрямку аерогідромеханіки, який визначається, як комп'ютерна гідродинаміка (CFD — Computational Fluid Dynamics).

Застосування розрахунків гідродинаміки складається з наступних етапів:

- На етапі *підготовка* утворюється геометрія моделі, формуються потрібні фізичні умови, дискретизується геометрія, задаються граничні і початкові умови для написання диференціальних рівнянь;
- *Розрахунок*. Машина, яка слідує заданим алгоритмам, вирішує основні рівняння з точки зору основних фізичних параметрів (тиск, швидкість, ентальпія, густина, температура і т. д.) чисельним методом, і записує результати в пам'ять;
- *Аналіз*. На цьому етапі результати рішення відображаються у формі таблиць чи графіків, векторних або контурних схем, які прив'язані до заданої геометрії.

За допомогою комп'ютерних експериментів, використовуючи сучасні технології вирішуються не лише інженерні проблеми в аеродинаміці, а і вирішується частина труднощів з проведенням дороговартісних експериментів, що проводяться в аеродинамічних трубах.

На сьогоднішній день, ринок пропонує значну кількість програмних продуктів, що використовуються для отримання характеристик потоку. Однією з таких програм являється комерційна програма скінчено-елементного аналізу ANSYS.

ANSYS користується популярністю серед спеціалістів в сфері інженерних розрахунків (Computer-Aided Engineering — CAE), рішення нелінійних і лінійних,

нестационарних і стаціонарних задач механіки у просторі зміненого твердого тіла та механіки конструкцій. ANSYS пропонує два програмних продукти – ANSYS CFX і ANSYS Fluent. На рисунку 2.2 показано приклад роботи вище вказаної програми, яка призначена для моделювання гідро-, аеродинамічних процесів, вирішуює ряд задач гідро-, газодинаміки, тепло масообміну, процесів горіння, та інших фізичних процесів, пов'язаних з фізикою рідини і газу.

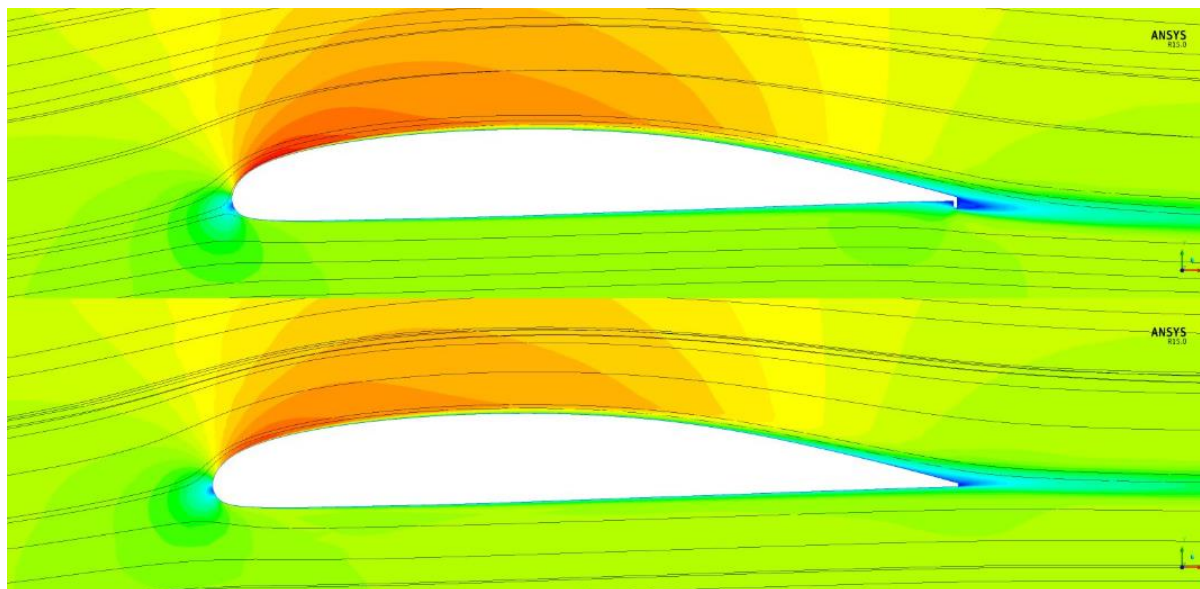


Рисунок 2.2 – Приклад візуалізації обтікання та аеродинамічних характеристик профілю крила за допомогою програмного комплексу ANSYS R15.0

Також даний продукт використовується для вирішення задач теплопередачі і теплообміну, електродинаміки, механіки рідини і газу, акустики та механіки зв'язаних полів. В деяких областях промисловості моделювання та аналіз дозволяє уникнути дорогі і тривалі цикли розробки виду «проектування - виготовлення - випробування».

2.3 Дослідження аеродинаміки крила

Сили тертя (дотичні до поверхні) та сили опору повітря (перпендикулярні до поверхні) при дії на літак чи будь-яку його частину створюють рівнодіючу аеродинамічну силу (рисунок 2.3). Зазвичай, цю рівнодіючу розглядають як геометричну суму:

$$R^2 = Y^2 + Q^2 \quad (2.7)$$

де, Y – підйимальною силою, Q – силою лобового опору.

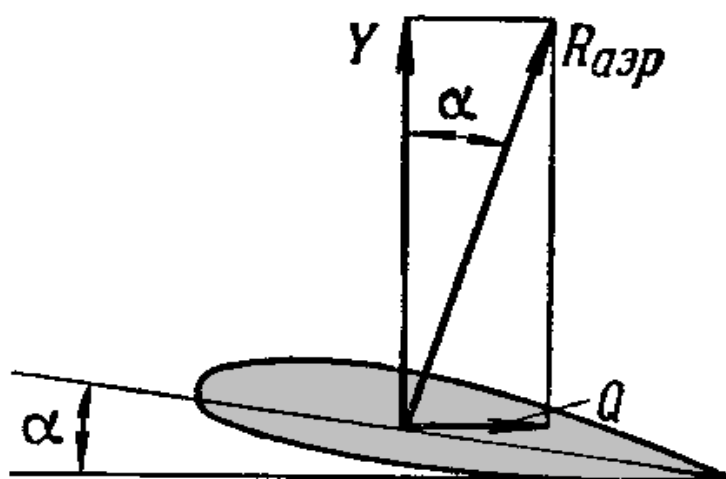


Рисунок 2.3 – Геометрична сума сили лобового опору і підйимальної сили

Ці сили можна розглядати як алгебраїчну суму наступних доданків: сили тертя і сили тиску. У підйимальній силі сили першим доданком можна знехтувати, вважаючи, що вона – тільки силою тиску. Опір розглядається як сума опору тертя і опору тиску.

Значення опору тиску і підйимальної сили залежить від розподілу тиску повітря по поверхні. В свою чергу розподіл тиску визначений характером обтікання потоком повітря літака.

2.3.1 Режими обтікання

На режим обтікання тіла впливає стан повітряного середовища, швидкість течії, форма тіла, що обтікається та його положення відносно потоку. В механіці суцільних середовищ Число Маха (М) – відношення швидкості течії в даній точці газового потоку до місцевої швидкості поширення звуку в рухомому середовищі:

$$M = \frac{V}{a}. \quad (2.8)$$

Основний потік біля тіла за характером руху може бути зривний та безвідривний (плавний). Знаючи число Маха основного повітряного потоку біля тіла можна розрізнити дозвукове, надзвукове та змішане обтікання. Місцеві числа М при дозвуковому обтіканні менші одиниці, а при надзвуковому – більші. При обтіканні змішаного характеру присутні області дозвукового і надзвукового потоку.

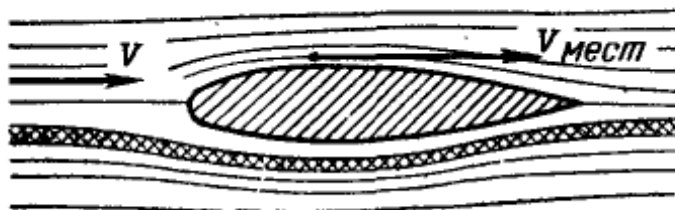
Докритичним обтікання називають при дозвукових швидкостях польоту. Змішане обтікання виникає Якщо дозвукові швидкості великі, а надзвукові ще малі, тоді виникає змішане обтікання крила. Така область швидкостей польоту називається колозвуквою або трансзвуквою. Польотні числа Маха для сучасних надзвукових літаків у трансзвуковій області швидкостей може бути в межах від 0,75 – 0,85 до 1,2 – 1,4.

При виникненні змішаного режиму обтікання, сукупність явищ, які спостерігаються називають хвильовою кризою. Вона виникає після перевищення критичних швидкостей польоту. Нижче, за умови безвідривного обтікання, розглянемо дозвуковий режим обтікання і розподілення тиску навколо нього.

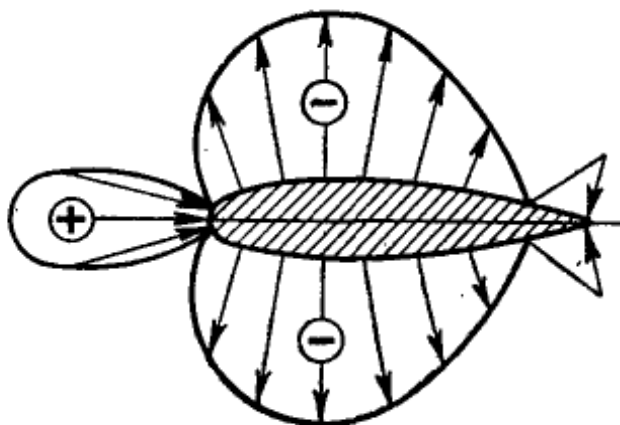
2.3.2 Дозвукове обтікання крила

На рисунку 2.4 зображена область обтікання крила на малих швидкостях (а) та розподіл тиску при цьому (б). Стрілки в певному масштабі показують величину надлишкового тиску:

В передній частині крила і за ним, швидкість потоку знижена, згідно закону Бернуллі надлишковий тиск додатний і найбільшу величину досягає в точці повного гальмування потоку – критична точка.



a



б

Рисунок 2.4 – Обтікання крила дозвуковим потоком

Верхня і нижня опуклості крила, навпаки, зменшують січення струєк: тут місцева швидкість перевищує швидкість незбудженого потоку, що створює від'ємний надлишковий тиск. Чим більше опуклість, тим сильніша розрідженість.

Коефіцієнт тиску в будь-якій точці залежить від відношення місцевої швидкості до швидкості набігаючого потоку, при умові якщо немає впливу стисливості і в'язкості при малих числах M і поза граничного слою,

При зміні форми, орієнтації відносно потоку (кут атаки) або обтікання тіла, не змінюючи швидкість V – місцеві швидкості в усіх точках зміняться, звідси, зміняться і коефіцієнти тиску. Але якщо, не змінити орієнтацію тіла, і не змінюючи форми збільшити швидкість потоку, то усі місцеві швидкості збільшаться пропорційно,

тобто відношення не зміниться, а отже і коефіцієнти тиску. Іншими словами, у такому випадку надлишковий тиск збільшиться прямопропорційно квадрату швидкості.

Коефіцієнти тиску не зміняться лише в тому випадку, якщо змінити розміри тіла, але при цьому зберегти геометричне співвідношення. Отже, рисунок розподілу коефіцієнтів тиску, отриманих для моделі, природна і для великого літака. Пропорційність порушується при малих числах M , коли помітний значний вплив стисливості газу. Наприклад, перед лобовою частиною крила, де потік пригальмований, густина повітря за рахунок стисливості підвищується, надлишковий тиск додатній, в свою чергу ущільнення повітря збільшує його опір на поверхню крила. І навпаки – зменшення густини дає можливість підвищити швидкість руху повітря, в точках, де місцева швидкість більша швидкості потоку газу, тому розрідженість стає більш сильною, чим при відсутності стисливості повітря. Отже, під впливом стисливості повітря розподілу тиску змінюється: абсолютні величини коефіцієнтів тиску підвищуються (рисунок 1.5).

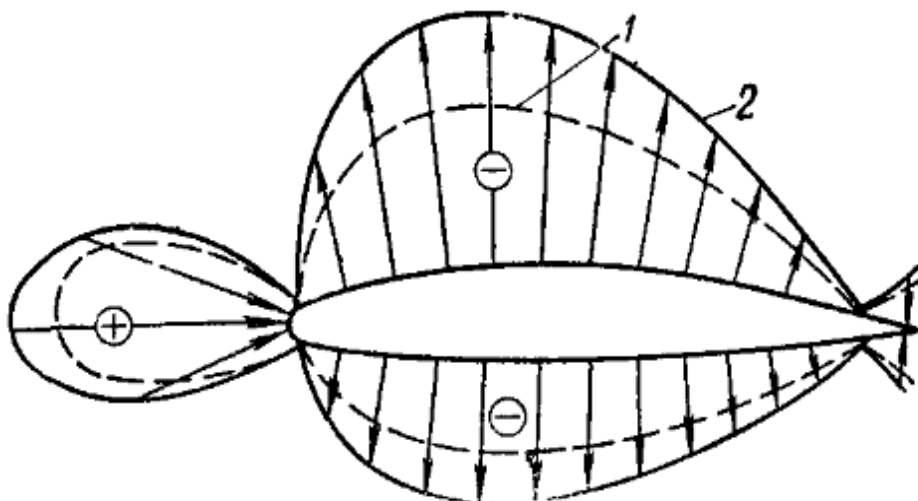


Рисунок 2.5 – Коефіцієнти тиску на поверхні крила при дозвуковому обтіканні: 1- слабкий прояв стисливості повітря при малих числах M ; 2 – значний прояв стисливості повітря при великих числах M

Зі збільшенням числа M , сильніше проявляється стисливість газу, це означає, що сильніше змінюються і коефіцієнти тиску. До поки швидкості польоту будуть до

критичними, така зміна має тільки кількісний характер. Перевищення критичного числа M є наслідком якісного зміння рисунку тиску.

2.3.3 Виникнення підйимальної сили.

Підйимальна сила перпендикулярна до вектора швидкості руху тіла в потоці газу або рідини та є складовою аеродинамічної сили. Вона створюється крилом в результаті різниці середніх тисків зверху і знизу крила. Підйимальна сила виникає через несиметричність обтікання тіла потоком. Її величина залежить від форми аеродинамічного профілю, в тому числі від степені його вигнутості (кривини, при якій верхня частина поверхні більш вигнута ніж нижня частина поверхні). В загальному випадку збільшення вигнутості збільшує підйимальну силу (рисунок 1.).



Рисунок 2,6. – Порівняння симетричного та вигнутого профілю

Вигнуті аеродинамічні профілі здатні утворювати підйимальну силу при нульовому куті атаки. Коли лінія хорди в горизонті, задня кромка крила направлена вниз і тому повітря, яке протікає повз задню кромку відхиляється вниз. Коли вигнутий профіль перевернутий навпаки, можна підібрати такий кут атаки, що підйимальна сила буде направлена вгору. Це пояснює як літак може здійснювати перевернутий політ. Це справедливо для дозвукового, надзвукового, і для змішаного обтікань. Крило симетричного профілю створює підйимальну силу і при куті атаки (α) відмінному від нуля. В режимі дозвукового обтікання підйимальна сила може утворюватися крилом при $\alpha=0$, при умові несиметричності профілю.

В результаті, швидкість потоку повітря під крилом менша, ніж над ним, що є наслідком більшої кривизни верхньої частини крила. Звідси, за законом Бернуллі тиск у верхній частині крила виявляється меншим, ніж знизу. При використанні

несиметричного профілю виникає додатковий потік і відбувається циркуляційний рух повітря, що зменшує середню швидкість потоку під крилом і збільшує її під ним. В наслідок цього виникає різниці тисків і утворюється підймальна сила. вплив циркуляційного руху на підймальну силу вперше був російським вченим Н.Е. Жуковським. На цій основі розроблена теорія крила, яка значно вплинула на розвиток авіації.

До навколишні умов середовища, що впливають на підймальну силу відносяться щільність, в'язкість і швидкість плинину. Підймальна сила пропорційна щільності середовища і приблизно пропорційна квадрату швидкості плинину. Щільність, в свою чергу, може залежати від температури і, при високих швидкостях що перевищують швидкість звука в середовищі, ефектами стискання. Підймальна сила також залежить від розміру крила, і в основному пропорційна від площі крила, що спроектована на напрям підймальної сили. В теорії аеродинаміки і інженерних розрахунках часто є зручним виразити підймальну силу кількісно через "коефіцієнт підймальної сили", що визначений таким чином, щоб врахувати ці пропорційні параметри.

Математичні теорії підймальної сили основані на механіці суцільних рідин, припускаючи що повітря плине як неперервна матерія. Підймальна сила утворюється в відповідності до фундаментальних законів фізики, найбільш актуальними є наступні три закони:

- збереження імпульсів, що в свою чергу є слідством Ньютонівських законів руху, особливо другий закон Н'ютона, який пов'язує силу прикладену до елемента повітря із зміною швидкості імпульсу.
- збереження мас, яка припускає, що поверхня профілю крила є непроникна для повітря, що перетікає довкола,
- збереження енергії, який говорить, що енергія не може бути створена чи знищена.

Коефіцієнт підймальної сили — безрозмірна величина, що характеризує підймальну силу крила певного профілю при відомому куті атаки. Коефіцієнт

визначається експериментально в аеродинамічній трубі, або за теоремою Жуковського. Формулу розрахунку підйимальної сили через коефіцієнт розробили брати Райт і Джон Смітон на початку XX століття. Формула має вигляд:

$$Y = C_y \frac{\rho V^2}{2} S \quad (2.9)$$

де:

Y — підйимальна сили (Н)

C_y — коефіцієнт підйимальної сили

ρ — густина повітря на висоті польоту (кг/м³)

V — швидкість потоку (м/с)

S — характерна площа (м²)

Для конкретного профілю крила, прийнято показувати відношення між коефіцієнтом підйомної сили і кутом атаки.

Коефіцієнт підйомної сили профілю оснований на двовимірному потоці довкола крила з нескінченним розмахом і незмінним поперечним перерізом, таким що підйомна сила не залежить від поздовжніх параметрів і визначається в термінах l , підйомна сила на одиницю розмаху крила. Тоді визначення буде наступним:

$$C_y = \frac{l}{\frac{1}{2} \rho v^2 c} \quad (2.10)$$

де c — хорда профілю крила.

Це повністю аналогічно для коефіцієнту аеродинамічного опору, оскільки хорду можна інтерпретувати як «поверхню на одиницю розмаху».

Для заданого кута атаки, C_y можна порахувати наближено з використанням теорії тонкого профілю, розрахувати чисельно або визначити за допомогою випробувань в аеродинамічній трубі, із тестовим зразку обмеженої довжини, із використанням торцевих пластин розроблених для покращення тривимірних ефектів. Графіки коефіцієнту C_y по відношенню до зміни кута атаки мають однакову загальну форму для всіх профілів, але конкретні числа будуть змінюватись. Вони показують майже лінійне збільшення коефіцієнту підйомної сили із збільшенням кута атаки із

деяким конкретним нахилом. Для тонкого профілю крила будь-якої форми нахил кривої коефіцієнту дорівнює $\pi^2/90 \simeq 0.11$ на градус. При збільшенні кута атаки буде досягнуте максимальне значення на кривій, після якого, коефіцієнт підйомної сили буде зменшуватись. Кут, на якому коефіцієнт підйомної сили буде максимальним, називається кутом звалювання профілю крила, який приблизно дорівнює від 10 до 15 градусів для типових профілів крила.

Для симетричних профілів також будують графіки для C_y на від'ємних кутах атаки, симетрично до осі C_y , крім того у будь-якого профілю із позитивним вигином, тобто асиметричним, опуклим зверху, буде невеликий, але позитивний коефіцієнт підйомної сили при кутах атаки нижче нуля. Це кут, який при $C_y = 0$ буде негативним. В таких профілях на нульових кутах атаки тиск на верхній поверхні профілю нижчий ніж на нижній поверхні.

Для прикладу: якщо $C_y=0.5$ то це означає, що на 1 м крила підймальна сила дорівнює половинні швидкісного опору.

Значення коефіцієнта підймальної сили залежить від:

1. форми крила та літака в цілому;
2. кута атаки (положення літака відносно потоку);
3. числа M .

$$C_y = \frac{0.07\alpha^\circ}{\sqrt{M^2-1}} \quad (2.11)$$

У випадку малої розмірності числа M можна вважати, що C_y показує вплив на величину підймальної сили тільки форми крила та кута атаки, і мало залежить від числа M . При трансзвукових швидкостях на величину C_y суттєво впливають всі три фактори, в той час коли при надзвуковому обтіканні при першому наближенні можна знехтувати впливом форми (форми крилового профілю) і рахувати C_y залежно від кута атаки та числа M згідно формули (2.11).

Розвиток зриву потоку можливий при великих кутах атаки, тоді величина C_y залежить в тому числі і від числа Рейнольдса, але ця залежність не буде розглядатись.

2.3.4 Виникнення лобового опору. Формула лобового опору

Сила лобового опору руху тіл в рідинах і газах. Лобове опір складається з двох типів сил: сил дотичного тертя, спрямованих уздовж поверхні тіла, і сил тиску, спрямованих по нормалі до поверхні. Сила опору є дисипативною силою і завжди спрямована проти вектора швидкості тіла в середовищі. Разом з підйнятною силою є складовою повної аеродинамічної сили.

Опір при нульовій підймальній силі не залежить від величини створюваної підйомної сили і складається з профільного опору крила, опору елементів конструкції літака, які не дають внеску в підйомну силу, і хвильового опору. Останній є істотним при русі з близько-і надзвуковою швидкістю, і викликаний утворенням ударної хвилі, яка забирає значну частку енергії руху. Хвильовий опір виникає при досягненні літаком швидкості, що відповідає критичному числу Маха, коли частина потоку, який оточує крило літака, набуває надзвукової швидкості. Критичне число M тим більше, чим більший кут стріловидності крила, чим більш загострена передня кромка крила і чим воно тонше. Сила опору спрямована проти швидкості руху, її величина записується наступним чином:

$$X_0 = C_{x0} \frac{\rho V^2}{2} S \quad (2.12)$$

C_{x0} — безрозмірний аеродинамічний коефіцієнт опору, отримується з критеріїв подібності, наприклад, чисел Рейнольдса і Фруда в аеродинаміці.

Потужність, необхідна для подолання даної складової сили лобового опору, пропорційна кубу швидкості.

У наслідок утворення підйомної сили на крилі скінченного розмаху утворюється **індуктивний опір**. Несиметричне обтікання крила призводить до того, що потік повітря збігає з крила під кутом до набігаючого на крило потоку. Таким чином, під час руху крила відбувається постійне прискорення маси набігаючого повітря в напрямку, перпендикулярному напрямку польоту, і направленому вниз. Це прискорення, по-перше, супроводжується утворенням підйомної сили, а по-друге — призводить до необхідності надавати кінетичну енергію потокові, що прискорюється.

Кількість кінетичної енергії, необхідна для надання потоку швидкості, перпендикулярної напрямку польоту, і визначатиме величину індуктивного опору. На величину індуктивного опору впливає не тільки величина підйомної сили, але і її розподіл за розмахом крила. Мінімальне значення індуктивного опору досягається при еліптичному розподілі підйомної сили за розмахом. При проектуванні крила цього домагаються наступними методами:

- вибором раціональної форми крила в плані;
- застосуванням геометричної та аеродинамічної скрутки;
- установкою допоміжних поверхонь — вертикальних кінцівки крила.

Індуктивний опір записується так:

$$X_i = C_{xi} \frac{\rho V^2}{2} S = \frac{C_y^2}{\pi \lambda} \frac{\rho V^2}{2} S = \frac{1}{\pi \lambda} \frac{Y^2}{\frac{\rho V^2}{2} S} \quad (2.13)$$

Таким чином, індуктивний опір вносить істотний внесок при польоті на малій швидкості (і, як наслідок, на великих кутах атаки). Він також збільшується при збільшенні ваги літака.

Сумарний опір є сумою всіх видів сил опору:

$$X = X_0 + X_i \quad (2.14)$$

Оскільки опір при нульовій підйомній силі X_0 пропорційний квадрату швидкості, а індуктивний X_i — обернено пропорційний квадрату швидкості, то вони вносять різний внесок при різних швидкостях. Із зростанням швидкості X_0 зростає, а X_i — падає, і графік залежності сумарного опору X від швидкості («крива потрібної тяги») має мінімум в точці перетину кривих X_0 і X_i , при якій обидві сили опору рівні за величиною. При цій швидкості літак має найменший опір при заданій підйомній силі (рівній вазі), а значить найвищу аеродинамічну якість.

Потужність, необхідна для подолання сили паразитного опору, пропорційна кубу швидкості, а потужність, необхідна для подолання індуктивного опору,

обернено пропорційна швидкості, тому сумарна потужність також залежність від швидкості нелінійно. При деякій швидкості потужність (а отже і витрата палива) стає мінімальною — це швидкість найбільшої тривалості польоту (баражування). Швидкість, при якій досягається мінімум відношення потужності (витрат палива) до швидкості польоту, є швидкістю максимальної дальності польоту або крейсерською швидкістю.

Якщо враховувати, що підйомна сила виникає тільки за рахунок крила то індуктивний опір можна розглядати як його опір, при чому інша частина опору крила (профільний опір) разом з шкідливим опором утворює без індуктивний опір. Відзначимо, що для будь якого опору для свого подолання необхідно витратити деяку роботи, тобто пов'язаний зв'язаний з втратою кінетичної енергії потоку, обтікаючи літак з переходом механічної енергії в теплову.

Розрахувати лобовий опір можна за формулами (1.25-1.26):

$$Q = C_x S \frac{\rho V^2}{2} \quad (2.15)$$

$$Q = 0,7 C_x S P M^2. \quad (2.16)$$

де C_x – коефіцієнт лобового опору літака який приходить на кожен квадратний метр площі крила.

Отже опір підйомна сила і тиску являються складовими рівнодіючої сили тиску. Звідси коефіцієнт опору тиску залежить від форма літака, кут атаки, числа М), як і коефіцієнт підйомної сили.

2.3.5 Аеродинамічна якість крила

Існує декілька основних аеродинамічних характеристиками профілю, такі як: коефіцієнт лобового опору $C_{x\alpha}$, коефіцієнт підйомної сили $C_{y\alpha}$. Отримані експериментальним шляхом значення коефіцієнтів зображуються графіками, що характеризують залежність цих коефіцієнтів від кута атаки α .

Аеродинамічною якістю профілю крила називається відношення підйомної сили до сили лобового опору або відношення коефіцієнта підйомної сили до коефіцієнта лобового опору

$$K = \frac{Y_{\alpha}}{X_{\alpha}} = \frac{C_{y\alpha}}{C_{x\alpha}}. \quad (2.17)$$

Застосування залежності $C_{y\alpha} = f(\alpha)$ і $C_{x\alpha} = f(\alpha)$, для кожного значення α можна отримати значення аеродинамічної якості профілю крила і побудувати залежність $K = f(\alpha)$. Якщо при певному куті якість крила досягає максимального значення, його називають найвигіднішим кутом атаки.

2.4 Висновки до другого розділу

1. Дослідивши моделювання аеродинамічних процесів, можна зробити висновки, що розподіл тиску на поверхню крила та процес обтікання крилового профілю течією ідеального нестисливого газового потоку представляє собою складну аналітичну задачу. В більшості випадків розв'язання математичних задач зв'язане з труднощами і вимагають проведення експериментів, часто досить складного і дорогого. Розвиток комп'ютерних технологій дещо допоміг вирішити ряд проблем.

2. На основі визначеної мети роботи та аналізу предметної області, виявлення нових підходів щодо моделювання обтікання профілів та визначення якісних оцінок крилового профілю, сформульовані наступні завдання досліджень:

- провести аналіз існуючих методів моделювання крилових профілів;
- розробити алгоритми для моделювання процесу обтікання на малих швидкостях польоту;
- розробити програмний комплекс, який дозволяє розрахувати якість крила для заданої геометрії профілю

3. МОДЕЛЮВАННЯ ПРОЦЕСУ АЕРОДИНАМІЧНОГО ОБТІКАННЯ КРИЛА

3.1 Аналіз методів для вирішення задачі знаходження аеродинамічних характеристик

Для вирішення задачі знаходження аеродинамічних характеристик літальних апаратів різних видів можна використовувати наступні методи: експериментальний, аналітичний, чисельний та чисельно-аналітичний.

Експериментальні методи (до них можна віднести експерименти, що проводяться в аеродинамічних трубах та льотні випробування), безперечно, є інструментом точної перевірки адекватності математичних моделей розрахунку аеродинамічних характеристик, вони є засобом отримання найповніших аеродинамічних характеристик, за ними залишається вирішальне слово при створенні нових літаків. Проте головним недоліком цього методу при визначенні особливостей обтікання і аеродинамічних характеристик експериментальних об'єктів – потреба проведення самих експериментів, значних масштабних робіт з виготовлення точних моделей (для аеродинамічних труб), обробці та аналізу отриманих даних.

Основою будь-якого дослідження в області обчислювальної гідродинаміки є формулювання основних рівнянь потоків гідро-, газодинаміки, а саме:

- рівняння нерозривності;
- рівняння збереження імпульсу;
- рівняння збереження енергії;
- рівняння станів (для газів).

Рівняння збереження імпульсу може мати різний вигляд в залежності від наявності або відсутності тертя. Нав'є-Стокса (для потоків з наявністю тертя) або Ейлера (відсутність тертя). Залежно від умов завдання середовище може розглядатися як стиснуте або нестисливе. В останньому випадку рівняння значно спрощуються.

Вказані вище рівняння представляють собою основну модель течії середовища, яка залежно від особливостей розв'язуваної задачі може бути доповнена рівняннями для моделей турбулентності, перенесення речовин, хімічних реакцій, обліку багатозфазності, електромагнітних взаємодій і т. д.

Всі перераховані математичні формулювання є системою нелінійних диференціальних рівнянь другого порядку, що мають аналітичне рішення лише в дуже простих випадках, коли число Рейнольдса для завдання мало, а геометрія проста.

Аналітичні методи аеродинаміки використовувалися в основному для вирішення загальних питань. З їх допомогою отримані розв'язки дуже вузького класу задач для тіл найпростіших геометричних форм. Тому аналітичні методи не можуть широко застосовуватись для отримання аеродинамічних характеристик складних просторових компонент.

Статичні *методи числового моделювання* (методи Монте-Карло) широкого класу задач швидко розвинулись за рахунок стрімкого розвитку обчислювальної техніки. Проте даний підхід висуває надзвичайно високі вимоги до ресурсів ЕОМ і не знайшов широкого застосування для вирішення практичних задач. Збільшення швидкодії й обсягу пам'яті електронних обчислювальних машин (ЕОМ), удосконалення чисельних методів аеродинаміки привело до інтенсивного зростання робіт з використання чисельних експериментів при розв'язанні важливих практичних задач. Чисельний експеримент органічно зв'язав у єдине ціле фізичну сутність явища, його раціональну схему, включаючи математичний опис, чисельний метод розв'язання.

Загально прийнято, що нестационарні рівняння Нав'є-Стокса цілком описують турбулентну течію реального газу. Для широкого обсягу технологічних і природних процесів задачу можна вирішити чисельно в тому випадку, якщо похідні, які стоять в рівняннях, замінити на кінцеві різниці, створені на малих просторових і часових інтервалах. У разі моделювання реального процесу проводиться так звана дискретизація простору і часу, таким чином, що геометрія процесу розбивається на

розрахункові осередки, обрані особливим чином, а час процесу - на розрахункові тимчасові інтервали.

Чисельно-аналітичні методи розв'язку гідродинамічних задач з рухомими межами володіють істотною перевагою в порівнянні з іншими методами, який пов'язаний з можливістю контролювання виконання граничних умов на поверхні розділу. Ця процедура здійснюється за рахунок кількості доданків у скорочених рядах, які використовуються при описі рішень поставленого завдання: розміри і форма тіла, розподіл полів тиску, функції струму та інших полів в розглянутій течії.

Перевагою цих методів є відносна простота отриманого рішення, досить висока точність виконання граничних умов і невисокі вимоги до обчислювальних ресурсів. Зауважимо, тривалість обчислень для більшості чисельно-аналітичних рішень складає одиниці хвилин в залежності від застосовуваного методу і точності, яка пред'являється до рішення.

Основним недоліком чисельно-аналітичних методів є їх обмеженість застосування до діапазону параметрів гідродинамічних течій. Для їх обґрунтування необхідно проводити аналіз масштабів сил і параметрів течій при формуванні моделі гідродинамічного течії з рухомими межами. Істотне ускладнення моделі в більшості випадків призводить до значного ускладнення отримання наближеного чисельно-аналітичного рішення.

До числа основних чисельних методів розв'язання рівнянь Нав'є-Стокса відносяться:

- метод скінченних елементів;
- метод скінченних різниць, або метод сіток;
- метод граничних елементів.
- проекційні методи:
 - метод Гальоркіна;
 - метод Рітца;
 - метод дискретних особливостей.

Розглянемо деякі з вище перелічених методів далі.

3.1.1 Метод скінчених елементів

Метод скінчених елементів — це потужний метод, розроблений для чисельного розв'язування складних проблем у механіці конструкцій та інших математичних задач; числова техніка знаходження розв'язків інтегральних та диференціальних рівнянь у частинних похідних (ДРЧП). Процес розв'язання побудований або на повному усуненні диференціального рівняння для стаціонарних задач, або на розкладі ДРЧП в апроксимуючу систему звичайних диференціальних рівнянь, які потім розв'язуються використанням якої-небудь стандартної техніки, такої як метод Ейлера, Рунге-Кутти тощо.

Під час розв'язання часткових диференціальних рівнянь основною метою є створення рівності, що апроксимує досліджувану рівність, і є чисельно стабільною, тобто помилки у вхідних даних і проміжних обчисленнях не акумулюються і не спричиняють беззмістовних результатів.

Розглянемо дію методу на двох прикладах, на які можна екстраполювати основний метод. $P1$ є одновимірною проблемою:

$$P1 : \begin{cases} u' = f \text{ in } (0,1), \\ u(0) = u(1) = 0, \end{cases}$$

де f є заданою функцією, а u — невідома функція від x і u' є другою похідною функції u по змінній x .

Двовимірна проблема є відомою під назвою проблема Діріхле:

$$P2 : \begin{cases} u_{xx} + u_{yy} = f \text{ in } \Omega, \\ u = 0 \text{ on } \partial\Omega. \end{cases}$$

де Ω — відкрита зв'язна областю на площині (x, y) , з «гарною» границею, а u_{xx} і u_{yy} — другі похідні функції u по x , та по y відповідно.

Проблему $P1$ можна розв'язати обрахуванням первісних. Хоча цей метод розв'язування задач на граничних значеннях працює коли є тільки один просторовий вимір і не узагальнюється на багатовимірні задачі чи на задачі виду

$u + u' = f$. З цієї причини ми застосуємо метод скінченних елементів на $P1$ і опишемо його узагальнення на $P2$.

Пояснення відбуватиметься у два кроки, які віддзеркалюють два основних кроки для розв'язання задач на граничні значення, використовуючи метод скінченних елементів:

- на першому кроці ми маємо перетворити ЧДР у його варіаційну форму;
- другий крок полягає у дискретизації, де форма є дискретизованою на скінченно вимірному просторі.

Після другого кроку ми маємо конкретну формулу для великої, зате скінченно вимірної лінійної задачі, розв'язок якої буде приблизно розв'язувати початкове ЧДР.

3.1.2 Метод граничних елементів

Метод граничного елемента (метод потенціалу, метод граничних інтегральних рівнянь) – метод вирішення крайової задачі, в якому завдяки використанню формул Гріна, вона зводиться до інтегрального рівняння на кордоні розрахункової області (найчастіше до інтегрального рівняння Фредгольма другого роду).

Неоднорідне рівняння Фредгольма другого роду виглядає так:

$$f(t) = \varphi(t) - \mu \int_b^a K(t,s)\varphi(s)ds. \quad (3.1)$$

Задача полягає в тому, щоб маючи ядро $K(t,s)$ і функцію $f(t)$, знайти функцію $\varphi(t)$. При цьому існування рішення і його множинність залежить від числа μ , що має назву характеристичним числом (зворотнє йому називається власним). Стандартний підхід вирішення використовує поняття резольвенти; записане у вигляді ряду рішення відоме як ряд Ліувілля - Неймана

$$\varphi(s) = \mu \int_b^a K(t,s)\varphi(t)dt + f(s). \quad (3.2)$$

У 80-ті метод граничного елемента розглядався як можливий конкурент методу скінченних елементів. Основна перевага в порівнянні з методом скінченних елементів - точний розв'язок вихідного диференціального рівняння всередині

розрахункової області. У задачах з нескінченною границею метод граничного елемента має перевагу через легкий спосіб розрахунку.

В цілому, можна сказати, що в рамках традиційної постановки задачі Діріхле, Неймана (і відповідні їм теорії пружності) для гладкої кордону успішно вирішуються. Можна використовувати аналітичне інтегрування (не завжди це раціонально з точки зору витрат машинних ресурсів) і метод послідовних наближень рішення СЛАР (на основі модифікованих рівнянь), для доказу збіжності яких використовується теорія інтегральних рівнянь Фредгольма другого роду.

Через складність реалізації і обмеженою сфери застосування інтерес до методу зменшився. Принаймні, заміною методу скінченних елементів, як очікувалося, він не став. Існує велика кількість постановок, відмінних від традиційної. У тому числі в тих випадках, коли математична теорія відсутня, а рівняння записати можна. Наприклад, рішення на основі рівняння Фредгольма першого роду, для чого необхідно виробляти регулярність, інакше завдання є некоректно поставленою (при невеликій зміні правій частині, рішення змінюється значно). Змішаної задачі, де необхідно врахувати можливу появу необмеженої похідною шуканої функції поблизу точки зміни граничних умов навіть при гладкому кордоні. Узагальнення для кусково-гладкого кордону (в плоскому випадку) можна проводити за допомогою рівнянь для гладкої кордону, шляхом введення вагових функцій, отриманих на основі дослідження асимптотних рішень для клина.

3.1.3 Метод Гальоркіна

Метод Гальоркіна (метод Бубнова - Гальоркіна) – метод наближеного розв’язку крайової задачі для диференціального рівняння $L[\psi(x)] = 0$. Тут оператор L може містити приватні або повні похідні шуканої функції.

Першим кроком в реалізації методу Гальоркіна є вибір набору базисних функцій, які:

- задовольняють граничним умовам;
- в межі нескінченної кількості елементів базису утворюють повну систему.

Конкретний вид функцій визначається зі специфіки завдання і зручності роботи. Часто застосовуються тригонометричні функції, ортогональні поліноми (поліноми Лежандра, Чебишева, Ерміта і ін.).

Рішення представляється у вигляді розкладання по базису:

$$\psi(x) = \sum_{k=1}^n \alpha_k \varphi_k(x). \quad (3.3)$$

Потім наближене рішення підставляється в вихідне диференціальне рівняння, і обчислюється його нев'язка. Для однорідного рівняння:

$$L \left[\sum_{k=1}^n \alpha_k \varphi_k(x) \right] = N(x). \quad (3.4)$$

Для неоднорідного рівняння $L[u] = f(x)$ нев'язка матиме вигляд $N(x) = L[u] - f(x)$.

Далі висувається вимога ортогональності нев'язки до базисних функцій, тобто:

$$\int_a^b N(x) \varphi_k(x) dx = 0. \quad (3.5)$$

Звідси виходить однорідна система рівнянь для коефіцієнтів в розкладанні, і вдається приблизно знайти власні значення завдання.

Методу Гальоркіна давно застосовуються як для вирішення диференціальних рівнянь з приватними похідними, так і для формування основи методу скінченних елементів. Застосування методу до дослідження задач стійкості гідродинамічних течій було реалізовано Г. І. Петровим, який довів збіжність методу Гальоркіна для відшукування власних значень широкого класу рівнянь, включаючи рівняння для неконсервативних систем, такі, як наприклад рівняння коливань в в'язкій рідині.

У гідродинаміці найбільш ефективно метод Гальоркіна працює в задачах про конвекцію. Задачі, пов'язані з течіями, не відносяться до таких і збіжність методу при невдалому виборі базису може бути сильно ускладнений.

3.1.4 Метод Рітца

Метод Рітца – прямий метод знаходження приблизного рішення крайових задач варіаційного числення.

Метод передбачає вибір пробної функції, яка повинна мінімізувати певний функціонал, у вигляді суперпозиції відомих функцій, які задовольняють граничним умовам. При цьому задача зводиться до пошуку невідомих коефіцієнтів суперпозиції. Просторовий оператор на операційному рівнянні, який описує крайову задачу, повинен бути лінійним, симетричним і позитивно-визначеним.

Метод Рітца застосовується для вирішення завдань варіаційного обчислення прямим методом. За допомогою прямих методів вирішуються вихідні завдання по знаходженню функції в заданому класі, які доставляють екстремальне значення заданому функціоналу.

Основні положення методу Рітца:

- Завдання по знаходженню функції $u(x)$ повинна бути сформульована в варіаційній формі;
- Рішення повинно бути представлено у вигляді кінцевого лінійного ряду виду:

$$u(x) = \sum_{i=1}^N c_i \varphi_i + \varphi_0(x), \quad (3.6)$$

де c_i - коефіцієнти Рітца, φ_i, φ_0 - апроксимаційні функції;

- Коефіцієнти c_i знаходяться з умов мінімізації функціоналу.

3.1.5 Метод дискретних особливостей

Метод дискретних особливостей часто використовується для вирішення завдань гідродинаміки. Він заснований на заміні безперервного вихрового шару, що моделює поверхню контактного розриву, його дискретним аналогом. Метод виконується по аналогії між потенційним і циркуляційним течіями ідеальної нестисливої рідини. При цьому вводиться система точок колокації, котра визначає геометрію профілю крила в поперечному розрізі, і систему точкових вихрів

невизначеної інтенсивності. При цьому підбирається така система стаціонарних точкових вихрів, яка формує еквівалентну течію в області течії, що розглядається. В місцях наближення ідеального нестисливого газового потоку інтенсивність вихрів підбирають такою, щоб в визначеному наборі контрольних точок (точки колокації) виконувалась гранична умова непротікання рідини через тіло. В результаті формується система лінійних алгебраїчних рівнянь, розв'язок якої дозволяє визначити інтенсивність стаціонарних вихрів.

Розглянемо течію ідеального нестисливого газу, на вході якого задана швидкість течії U_0 . Необхідно визначити розподілення поля швидкості в течії, що розглядається.

Відомо, що функція току $\Psi(x, y)$ течії газу, що наведена одиничним точковим вихрем інтенсивності Γ , розміщеним в точці з координатами (x_V, y_V) , визначається виразом

$$\Psi(x, y) = -\frac{\Gamma}{4\pi} \ln[(x_V - x)^2 + (y_V - y)^2]. \quad (3.7)$$

Якщо в систему, що розглядається входять N точкових вихрів з інтенсивністю Γ_i , $(i = 1, \dots, N)$, тоді вираз для функції току представляє собою суперпозицію вкладів кожного з вихрів

$$\Psi(x, y) = -\frac{1}{4\pi} \sum_{i=1}^N [(x_{Vi} - x)^2 + (y_{Vi} - y)^2], \quad (3.8)$$

де (x_i, y_i) – координати i -го точкового вихру.

Використовуючи зв'язок між функцією току та полем швидкості течії

$$U(x, y) = \frac{\partial \Psi}{\partial y}, \quad V(x, y) = \frac{\partial \Psi}{\partial x} \quad (3.9)$$

можна визначити значення проекцій швидкості течії в довільній точці.

Для того, щоб задовольнити граничні умови непротікання газу в термінах функції току

$$\psi(x, y)|_s = \text{const},$$

введемо систему N точок колокації та будемо вимагати, щоб в цих точках значення функції току, приведеної системою N точкових вихрів були однакові.

Накладаючи умови рівності значень функції току в точках колокації, отримуємо систему лінійних алгебраїчних рівнянь відносно невідомих інтенсивностей Γ_i точкових вихрів

$$\begin{aligned} |A_{ij}| &= \Psi_j, & i, j &= 1, \dots, N, \\ |A_{ij}| &= -\frac{1}{4\pi} \ln[(x_V - x)^2 + (y_V - y)^2]. \end{aligned} \quad (3.10)$$

де Ψ_j – значення функції току в j – й точці колокації.

Зазвичай несучий вихровий шар рівномірно розбивається на елементи, кожен з яких замінюється одним дискретним вихором, а контрольні точки розташовуються на рівних відстанях між вихорами. Така розрахункова схема забезпечує збіжність рішення для внутрішньої частини несучого вихрового шару, тоді як поблизу його межі може виникнути невинна похибка.

Відомо, що зменшення зміщення Δ призводить к погіршанню виконання граничної умови, особливо в області, рівновіддаленій від точок колокації. Присвоєння Δ великого значення по зрівнянню з відстанями між точками колокації призводить до того, що система алгебраїчних рівнянь (3.10) стає виродженою та чисельний її розв'язок стає проблематичним та вимагає застосування спеціальних методів розв'язку. Існує оптимальне значення Δ для заданої системи точок колокації, яке може бути визначене з умови мінімуму числа обумовленості алгебраїчної системи рівнянь (3.10).

В чисельних методах під числом C обумовленості розуміють число, яке дозволяє оцінити точність чисельного розв'язку лінійної алгебраїчної системи рівнянь (3.10).

В залежності від точності представлення вектора правих частин. Для будь-якої не виродженої матриці $[A]$ число обумовленості для системи лінійних алгебраїчних рівнянь виду

$$[A]x = b \quad (3.11)$$

представляє собою вираз

$$C = \text{Cond}[A] = \|A\| \cdot \|A^{-1}\|. \quad (3.12)$$

Число обумовленості визначає нерівність

$$\left| \frac{\delta x}{x} \right| \leq \|A\| \cdot \|A^{-1}\| \cdot \left| \frac{\delta b}{b} \right|. \quad (3.13)$$

Якщо в наведеній формулі відношення $\frac{|\delta b|}{|b|}$ інтерпретувати як міру відносної невизначеності вектору правих частин (для дійсних чисел з подвійною точністю маємо $\approx (10^{-20} \dots 10^{-16})$), тоді добуток $\text{Cond}[A] \cdot \frac{|\delta b|}{|b|}$ визначає обмеження зверху відносної невизначеності вектора розв'язку $\frac{|\delta x|}{|x|}$.

За допомогою значень Γ_i , де $i=1, \dots, N$, які отримали після розв'язку системи (3.7), можна визначити розподіл функції току $\Psi(x, y)$ навколо тіла по формулі (3.8).

В результаті можна зробити висновок, що для розрахунку аеродинамічного обтікання крилового профілю в момент наближенні потенційної течії ідеального газу, найбільш ефективно використовувати метод дискретних особливостей, який дозволяє для практичних цілей з достатньою точністю отримувати результати розрахунків. Аналізуючи результати розрахунків бачимо, що по мірі ускладнення математичної моделі можна отримати більш широкий спектр аеродинамічних характеристик, але в більшості випадках розв'язання відповідних математичних задач, які пов'язані з великими труднощами і дані про конкретні випадки обтікання тіл одержують досить складно і довго.

3.2 Використання методів обчислень для розв'язку математичної задачі процесу моделювання обтікання крилового профілю

Особливість завдання яке потрібно виконати допускає спрощення початкової задачі, проігнорувавши певними явищами (без в'язкості, політ у докритичному

режимі). З вище описаного знаємо, задачу аеродинамічного обтікання крила краще вирішується методом дискретних особливостей, що зведена до задачі Коші і може бути розв'язана за допомогою чисельних методів. Це дозволяє зменшити час виконання обчислень, їх кількість та скоротити процес реалізації.

Розглянемо специфіку вирішення завдань та методи їх розв'язання.

3.2.1 Принцип розміщення точок колокації

Для реалізації методів розв'язання потрібно вибирати контрольні точки, враховуючи локальні особливості вихрового шару. В їх якості вибираються точки, в яких розглянута система дискретних вихорів та ініційовані ті ж швидкості, що й у вихідному вихровому шарі. Даний вибір контрольних точок забезпечує збіжність наближеного розв'язку до точного у всій області, включаючи межі крила.

В чисельному методі дискретних особливостей для збереження гладкості розподілу поля швидкості біля границі течії пропонується змістити положення точкових вихрів на відстані Δ всередину крилового профілю, як показано на рисунку 3.1. Фіксовані вихорі позначаються кружечками, а контрольні точки зображені квадратиками. Кожний вихор розміщується на прямій AD перпендикулярно до дотичної BC з межею течії, на відстані Δ від точки колокації A .

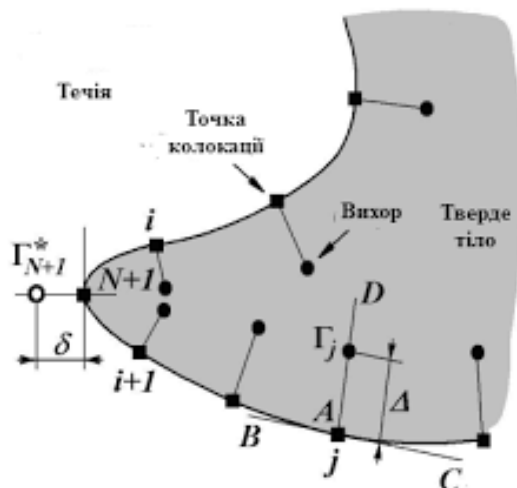


Рисунок 3.1 – Схема розташування точкових вихрів та точок колокації

Використовуючи метод дискретних особливостей при проведенні розрахунків важливим є вибір просторового розташування точкових вихрів та точок колокації. Правильний вибір – успіх вирішення задачі. Кількість точок колокації залежить від складності геометрії крилового профілю (рисунок 3.2).

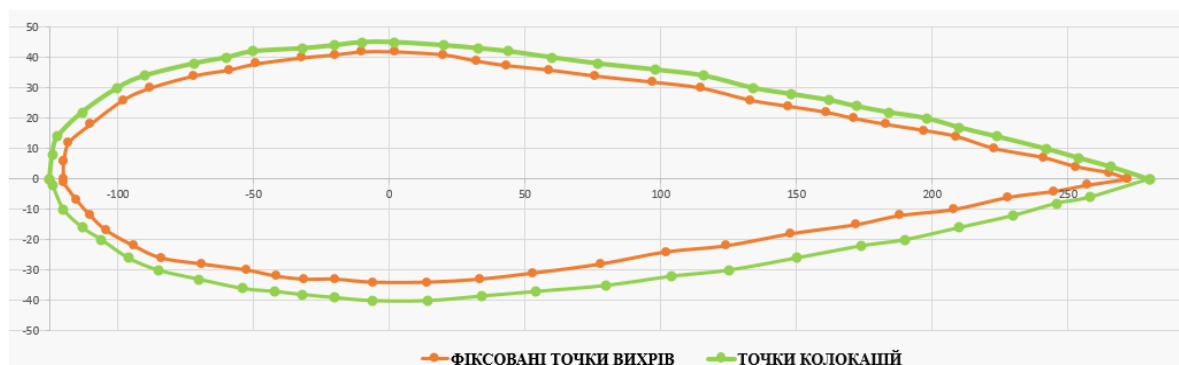


Рисунок 3.2 – Розташування систем точок вихрів і колокації на прикладі крилового профілю NASA – 0012

3.2.2 Метод Гауса для СЛАР

Важливу роль у математиці відіграють системи лінійних алгебраїчних рівнянь. До них зводяться теорії диференціальних рівнянь, багато задач лінійної алгебри, математичної фізики тощо, а також області техніки та фізики, в яких застосовуються ці математичні теорії.

Методи розв’язування СЛАР діляться на три групи: ітераційні, точні та ймовірнісні. За Бахваловим (1987 рік), до систем з числом змінних до порядку 10^4 застосовуються точні методи, 10^7 — ітераційні методи.

Метод послідовного виключення змінних (або метод Гауса) являється найпростішим методом розв’язання СЛАР. Є декілька модифікацій цього методу. Систему розв’язують у два етапи за рахунок схеми єдиного ділення. Спочатку вихідна система зводиться до трикутної форми. Таке перетворення називають прямим ходом. Другий етап має назву зворотній хід. На ньому знаходять розв’язок лінійної системи рівнянь трикутної форми.

Суть алгоритму розглянемо на прикладі системи, яка складається з трьох лінійних алгебраїчних рівнянь з трьома невідомими (визначник системи не дорівнює нулю):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = a_{14}, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = a_{24}, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = a_{34}, \end{cases} \quad (3.14)$$

Нехай $a_{11} \neq 0$ (цього можна досягти шляхом перестановки рівнянь системи, а оскільки визначник не дорівнює нулю, завжди є таке рівняння, в якому $a_{1i} \neq 0$). Ділимо коефіцієнти першого рівняння системи (3.14), і вільний член, на коефіцієнт a_{11} . Отримали нове рівняння

$$x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 = a_{14}^{(1)}, \quad (3.15)$$

де

$$a_{1j}^{(1)} = a_{1j}/a_{11}, \quad (j=2,3,4). \quad (3.16)$$

Позбавимось змінної x_1 з другого і третього рівнянь системи (2.14). Для цього, спочатку послідовно помножимо рівняння (2.15) на коефіцієнт a_{21} і віднімемо його від другого рівняння системи (2.14), а потім на a_{31} і віднімемо від третього рівняння системи (2.14). Отримали систему двох рівнянь зі змінними x_2 та x_3

$$\begin{cases} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = a_{24}^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = a_{34}^{(1)}, \end{cases} \quad (3.17)$$

де коефіцієнти a_{ij} обчислюють за формулами

$$a_{ij}^{(1)} = a_{ij} - a_{i1}^{(1)}a_{1j} \quad (i=2,3; j=2,3,4). \quad (3.18)$$

На те, що над коефіцієнтами системи (3.17) виконано перше перетворення, вказує верхній індекс.

На наступному кроці потрібно поділити коефіцієнти першого рівняння системи (2.17) на $a_{22}^{(1)}$ (якщо $a_{22}^{(1)} \neq 0$, переставляємо рівняння місцями).

Отримаємо рівняння:

$$x_2 + a_{23}^{(2)} = a_{24}^{(2)}, \quad (3.19)$$

де

$$a_{2j}^{(2)} = a_{2j}^{(1)} / a_{22}^{(1)}, \quad (j = 3, 4). \quad (3.20)$$

Аналогічно із системи (3.17) виключимо змінну x_2 так само, як x_1 з системи (3.14). Отримаємо рівняння:

$$a_{33}^{(2)} x_3 = a_{34}^{(2)}, \quad (3.21)$$

де

$$a_{3j}^{(2)} = a_{3j}^{(1)} - a_{32}^{(1)} a_{2j}^{(2)}, \quad (j = 3, 4). \quad (3.22)$$

З рівняння (3.21) маємо:

$$x_3 = a_{34}^{(3)}, \quad (3.23)$$

де

$$a_{34}^{(3)} = a_{34}^{(2)} / a_{33}^{(2)}. \quad (3.24)$$

В результаті отримаємо систему рівнянь трикутної форми після трьох кроків перетворення:

$$\begin{cases} x_1 + a^{(1)}_{12}x_2 + a^{(1)}_{13}x_3 = a^{(1)}_{14}, \\ x_2 + a^{(2)}_{23}x_3 = a^{(2)}_{24}, \\ x_3 = a^{(3)}_{34}, \end{cases} \quad (3.25)$$

яка тотожна системі (3.14). Цим завершується перший етап методу Гауса. Вище наведений процес перетворень системи (3.14) до еквівалентної їй системи (3.25) можна здійснити при виконанні наступних умов: $a_{11} \neq 0$, $a_{22}^{(1)} \neq 0$, $a_{33}^{(2)} \neq 0$.

У випадку, коли в системі (3.14) діагональні коефіцієнти близькі до нуля, може відбутися значна втрата точності.

Зважаючи на те, що системи (3.14) і (3.25) еквівалентні, розв'язком системи (3.14) буде розв'язок системи (3.25), який можна записати формулами:

$$\begin{cases} x_3 = a^{(3)}_{34}, \\ x_2 = a^{(2)}_{24} - a^{(1)}_{23}x_3, \\ x_1 = a^{(1)}_{14} - a^{(1)}_{13}x_3 - a^{(1)}_{12}x_2. \end{cases} \quad (3.26)$$

Зворотній хід методу Гауса завершено. Щоб виконати перетворення не обов'язково виписувати всю систему, можна обмежитись лише матрицею коефіцієнтів при змінних та вільними членами, бо перетворення систем лінійних рівнянь є фактично зміною їх коефіцієнтів.

3.2.3 Метод Рунге-Кутта-Фельберга 4-го порядку

При математичному моделюванні сучасних технічних пристроїв виникають складні складові системи нелінійних диференціальних рівнянь. Дослідження функціонування таких систем не може обійтись без рішення отриманих рівнянь математичної моделі. Якщо диференціальні рівняння, що описують роботу таких систем, є нелінійним і нестационарними, то часто неможливо отримати їх аналітичне рішення. Виникає необхідність використовувати чисельні методи, найбільш відомим з яких є метод Рунге – Кутта - Фельберга 4-го порядку. При цьому якщо математична модель складається з декількох систем диференціальних рівнянь, що описують окремі

ділянки функціонування, то виникає задача визначення моментів переходу від однієї системи до іншої. Для моделі, заданої складовими системами диференціальних рівнянь, умови закінчення одного етапу служать початковими значеннями для іншого. При чисельному рішенні неминуче виникає похибка, пов'язана як з похибкою самого методу, так і з похибкою початкових умов. Похибки, що виникають при моделюванні функціонування таких систем, можуть сильно впливати на підсумкове рішення і викликати суттєві відхилення від значень процесу реального об'єкта.

У зв'язку з цим виникає необхідність використовувати ефективні чисельні алгоритми розв'язання систем диференціальних рівнянь, що дозволяють швидко і з контрольованою похибкою знаходити рішення.

Розглянемо задачу Коші для системи диференціальних рівнянь довільного порядку, які записується у векторній формі:

$$y' = f(x, y), \quad y(x_0) = y_0. \quad (3.27)$$

Після цього значення невідомої функції в точці x_{n+1} рахуємо відносно значення в попередній точці x_n наступним чином за формулами

$$y_{n+1} = y_n + \frac{h}{6(k_1 + 2k_2 + 2k_3 + k_4)}, \quad (3.28)$$

$$x_{n+1} = x_n + h,$$

де h – заданий крок інтегрування, а коефіцієнти k_n розраховуються за формулами (3.29 – 3.32):

$$k_1 = f(x_n, y_n), \quad (3.29)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \quad (3.30)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \quad (3.31)$$

$$k_4 = f(x_n + h, y_n + hk_3). \quad (3.32)$$

У методі 4-го порядку, похибка на кожному кроці дорівнює $O(h^5)$, тоді сумарна похибка на останньому інтервалі інтегрування рівна $O(h^4)$.

3.3. Висновки до третього розділу

1. Щоб визначити можливості розв'язання задачі, виявлення нових методів реалізації комп'ютерного моделювання обтікання крилового профілю в нескінченному середовищі проаналізовано існуючі методи, а саме:

- метод граничних елементів та метод скінченних елементів, які є недостатньо точними при виконанні граничних умов на обмежуючій поверхні. Ці методи не є ефективними, оскільки відбувається протікання під час процесу переносу рідини в областях, що прилягають до границь;

- чисельно-аналітичні методи розв'язку гідродинамічних задач зі змінними границями. Їхніми перевагами є можливість контролювати процес виконання граничних умов на обраній поверхні, але дані методи обмежені в діапазоні параметрів гідродинамічних течій.

2. Провівши аналіз методів рішення задачі, виявлено, що для розрахунку обтікання крилового профілю доцільно використовувати модель потенційної течії ідеального газу. Вона дозволяє з достатньою точністю отримувати результати розрахунків обтікання просторових об'єктів для практичних цілей.

3. З аналізу переваг та недоліків розглянутих методів аргументовано необхідність застосування методу дискретних особливостей, який зводиться до вирішення задачі Коші та диференціальних рівнянь методом Рунге-Кутта 4-го порядку.

4. ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для вирішення задачі моделювання аеродинаміки профілю крила зазвичай використовують програмні комплекси ANSYS та FlowVision.

Вище згадані пакети прикладних програм відносяться до класу пропріетарного програмного забезпечення та має високу вартість. Придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: заборонений або закритий доступ до вивчення коду, внесення змін, розповсюдження та інші.

Програмний продукт моделювання аеродинамічної якості крилового профілю в нескінченній рідині містить простий та інтуїтивно зрозумілий користувачу інтерфейс. Містить позбавлений від надмірного перевантаження панель керування вхідними та вихідними даними. Потребує лише розуміння фізики процесу, відокремлюючи тим самим користувача від складних математичних обчислень, які виконує програма. Крім того програмний код є відкритим і доступним для вивчення, а використання програмного пакету – безкоштовним.

Таким чином були відмічені переваги використання даного програмного продукту перед вже існуючими системами розробки та моделювання математичних задач. В даному розділі наведені основні технології та кроки по програмній реалізації розроблених методів.

4.1 Мова програмування Python

Python— інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб

поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

4.1.1 Об'єктно-орієнтоване програмування

Дизайн мови Python побудований навколо об'єктно-орієнтованої моделі програмування. Реалізація ООП в Python є елегантною, потужною та добре продуманою, але разом з тим, достатньо специфічною в порівнянні з іншими об'єктно-орієнтованими мовами.

Можливості та особливості:

1. Класи є одночасно об'єктами з усіма нижче наведеними можливостями.
2. Успадкування, в тому числі множинне.
3. Поліморфізм (всі функції віртуальні).
4. Інкапсуляція (два рівні — загальнодоступні та приховані методи і поля). Особливість — приховані члени доступні для використання та помічені як приховані лише особливими іменами.
5. Спеціальні методи, що керують життєвим циклом об'єкта: конструктори, деструктори, розподільники пам'яті.
6. Перевантаження операторів (усіх, крім is, '!', '=' і символічних логічних).
7. Властивості (імітація поля за допомогою функцій).
8. Управління доступу до полів (емуляція полів і методів, частковий доступ тощо).
9. Методи для управління найпоширенішими операціями (істинносне значення, len(), глибоке копіювання, серіалізація, ітерація по об'єкту)
10. Метапрограмування (управління створенням класів, тригери на створення класів, та ін)
11. Повна інтроспекція.

12.Класові та статичні методи, класові поля.

13.Класи, вкладені у функції та інші класи.

4.1.2 Функціональне програмування

Python підтримує парадигму функціонального програмування, зокрема:

- функція є об'єктом;
- функції вищих порядків;
- рекурсія;
- розвинена обробка списків (спискові вирази, операції над послідовностями, ітератори);
- аналог замикань (closures);
- часткове застосування функції;
- можливість реалізації інших засобів на самій мові

4.1.3 Модулі та пакети

Програмне забезпечення (застосунок або бібліотека) на Python оформлюється у вигляді модулів, які у свою чергу можуть бути зібрані в пакунки. Модулі можуть розташовуватися як у каталогах, так і в ZIP-архівах. Модулі можуть бути двох типів за своїм походженням: модулі, написані на «чистому» Python, і модулі розширення (extension modules), написані на інших мовах програмування. Наприклад, в стандартній бібліотеці є «чистий» модуль pickle і його аналог на Cі: cPickle. Модуль оформляється у вигляді окремого файлу, а пакет — у вигляді окремого каталогу. Підключення модуля до програми здійснюється оператором `import`. Після імпорту модуль представлений окремим об'єктом, що дає доступ до простору імен модуля. У ході виконання програми модуль можна перезавантажити функцією `reload()`.

4.1.4 Інтроспекція

Python підтримує повну інтроспекцію часу виконання. Це означає, що для будь-якого об'єкта можна отримати всю інформацію про його внутрішню структуру.

Застосування інтроспекції (метапрограмування) є важливою частиною того, що називають «pythonic style», і широко застосовується в бібліотеках і фреймворках Python, таких як PyRO, Pyro, PLY, CherryPy, Django та інших, заощаджуючи час програміста, що ними користується.

4.1.5 Ітератори

У програмах на Python широко використовуються ітератори. Цикл `for` може працювати як з послідовністю, так і з ітераторами. Усі колекції, як правило, надають ітератор. Об'єкти визначеного користувачем класу теж можуть бути ітераторами. Модуль `itertools` стандартної бібліотеки містить багато корисних функцій для роботи з ітераторами. На відміну від звичайних послідовностей, всі елементи яких зберігаються в пам'яті, отримання наступного елемента забезпечує генератор — спеціальна функція, звернення до якої обчислює і повертає наступний елемент генератора.

4.1.6 Генератори

Однією з цікавих можливостей мови є генератори — функції, що між викликами зберігають внутрішній стан: значення локальних змінних і поточну інструкцію (див. також: супрограма). Генератори можуть використовуватися як ітератори для структур даних і для лінивих обчислень. Наприклад генератор чисел Фібоначчі:

```
def fib():  
    a, b = 1, 1  
    while True:  
        yield a  
        a, b = b, a + b
```

```
while x in fib():
    print(x)
```

При виклику генератора функція негайно повертає об'єкт-ітератор, який зберігає поточну точку виконання та стан локальних змінних функції. При запиті наступного значення (за допомогою методу `next()`, який неявно викликається в циклі `for`) генератор продовжує виконання функції від попередньої точки зупину до наступного оператора `yield` або `return`.

У Python 2.4 з'явилися генераторні вирази — вирази, що дають у результаті генератор. Генераторні вирази дозволяють заощадити пам'ять там, де інакше потрібно було б використовувати список із проміжними результатами:

```
>>> sum(i for i in xrange(1, 100) if i % 2 != 0)
2500
```

У цьому прикладі підсумовуються всі непарні числа від 1 до 99.

Починаючи з версії 2.5, Python підтримує повноцінні співпроцедури: тепер в генератор можна передавати значення за допомогою методу `send()` та збуджувати в його контексті винятки за допомогою методу `throw()`.

4.1.7 Керування контекстом виконання

У Python 2.5 з'явилися засоби для керування контекстом виконання блоку коду — оператор `with` та модуль `contextlib`.

Оператор може застосовуватися в тих випадках, коли до та після деяких дій обов'язково мають виконуватися якісь інші дії, незалежно від створених у блоці винятків або операторів `return`: файли має бути закрито, ресурси звільнено, перенаправлення стандартного введення/виведення завершено, тощо. Оператор полегшує читання коду, отже, допомагає уникати помилок.

4.1.8 Декоратори

Починаючи з версії 2.4, Python дозволяє використовувати, так звані, Декоратори (не слід плутати з однойменним шаблоном проектування) для підтримки

існуючої практики перетворення функцій та методів у місці визначення (декораторів може бути декілька). Після довгих дебатів для декораторів став використовуватися символ `@` у рядках, що передують визначенню функції або методу. Наступний приклад містить опис статичного методу без застосування декоратора:

```
def myWonderfulMethod ():
    return "Деякий метод"
myWonderfulMethod = staticmethod (myWonderfulMethod)
```

і за допомогою декоратора:

```
@staticmethod
def myWonderfulMethod ():
    return "Деякий метод"
```

Декоратор — це функція, першим аргументом якої є декорована функція або метод. Декоратори можна вважати елементом аспектно-орієнтованого програмування.

З версії 2.6 декоратори можна використовувати з класами, аналогічно функціям.

4.1.9 Бібліотеки

Багата **стандартна бібліотека** є однією з привабливих сторін Python. Тут є засоби для роботи з багатьма мережевими протоколами та форматами Інтернету, наприклад, модулі для написання HTTP-серверів та клієнтів, для розбору та створення поштових повідомлень, для роботи з XML, тощо. Набір модулів для роботи з операційною системою дозволяє писати крос-платформні застосунки. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізацією даних, юніт-тестуванням та ін.

Крім стандартної бібліотеки існує багато інших, що надають інтерфейс до всіх системних викликів на різних платформах; зокрема, на платформі Win32 підтримуються всі виклики Win32 API, а також COM в обсязі не меншому, ніж у Visual Basic або Delphi. Існує велика кількість прикладних бібліотек для Python у

різноманітних галузях: веб-розробка, бази даних, обробка зображень, обробка тексту, чисельні методи, програми операційної системи тощо.

Для Python прийнята специфікація програмного інтерфейсу до баз даних DB-API 2 та розроблено відповідні цій специфікації пакети для доступу до різних СУБД: PostgreSQL, Oracle, Sybase, Firebird (Interbase), Informix, Microsoft SQL Server, MySQL та sqlite. На платформі Microsoft Windows доступ до БД можливий через ADO (ADODB). Комерційний пакет mxODBC для доступу до СУБД через ODBC для платформ Windows і UNIX розроблений eGenix. Для Python написано багато ORM: (SQLObject, SQLAlchemy, Dejavu, Django), виконані програмні каркаси для розробки веб-застосунків (Django, Pylons).

Бібліотека NumPy для роботи з багатовимірними масивами дозволяє досягти продуктивності наукових розрахунків, порівнянної зі спеціалізованими пакетами. SciPy використовує NumPy і надає доступ до великого спектру математичних алгоритмів (матрична алгебра — BLAS, level 1-3 і LAPACK;).

Бібліотека WSGI — інтерфейс шлюзу з веб-сервером (Python Web Server Gateway Interface).

Python надає простий і зручний програмний інтерфейс C API для написання власних модулів на мовах C та C++. Інструмент SWIG дозволяє майже автоматично отримувати прив'язки для використання C/C++ бібліотек у коді на Python. Можливості цього та інших інструментів варіюються від автоматичної генерації (C/C++/Fortran)-Python інтерфейсів за спеціальними файлами (SWIG, pyste, SIP, pyfort) до надання зручніших API (boost::python, CXX та ін.) Інструмент стандартної бібліотеки ctypes дозволяє програмам Python безпосередньо викликати функції з динамічних бібліотек/DLL, написаних на C. Існують модулі, що дозволяють вбудовувати код на C/C++ прямо у вихідні файли Python, створюючи розширення «на льоту» (pyinline, weave). Для підключення математичних функцій, особливо із застосуванням NumPy, наразі офіційно рекомендованим є Cython

Інший підхід полягає у вбудовуванні інтерпретатора Python у застосунки. Python легко вбудовується в програми на Java, C/C++, Ocaml. Взаємодія Python-застосунків з іншими системами можлива також за допомогою CORBA, XML-RPC, SOAP, COM.

За допомогою Pyrex можлива компіляція Python-подібної мови (додано можливість типізації) в еквівалентний Сі-код і зв'язування із зовнішніми модулями.

Експериментальний проект shed skin передбачає створення компілятора для трансформації неявно типізованих Python програм в оптимізований C++ код. Починаючи з версії 0.22 shed skin дозволяє компілювати окремі функції в модулі розширень. Повна компіляція (станом на 1 липня 2007) далека від завершення.

Python та переважна більшість бібліотек до нього безкоштовні й поставляються у вихідних кодах. Навіть більше, на відміну від багатьох відкритих систем, ліцензія ніяк не обмежує використання Python у комерційних розробках та не накладає ніяких зобов'язань, крім зазначення авторських прав.

Використовуються також графічні бібліотеки.

З Python поставляється бібліотека tkinter на основі Tcl/Tk для створення крос-платформних програм з графічним інтерфейсом. Для науково-технічної мети найбільшого поширення набуло використання matplotlib — бібліотеки з інтерфейсом, аналогічним MATLAB Plot Tool.

Існують розширення, що дозволяють використовувати всі основні GUI бібліотеки — wxPython, засноване на бібліотеці wxWidgets, PyGTK для GTK+, PyQt та PySide для Qt та інші. Деякі з них також надають широкі можливості для роботи з базами даних, графікою та мережами, використовуючи всі можливості бібліотеки, на якій базуються.

Для створення ігор та програм, що вимагають нестандартного інтерфейсу, можна використовувати бібліотеку Pygame. Вона також надає великі засоби роботи з мультимедіа: з її допомогою можна керувати звуком і зображеннями, відтворювати відео. Надаване pygame апаратне прискорення графіки OpenGL має більш високорівневий інтерфейс в порівнянні з PyOpenGL, що копіює семантику C-бібліотеки для OpenGL. Є також PyOgr, що забезпечує прив'язку до OGRE — високорівневої об'єктно-орієнтованої бібліотеки 3D-графіки. Крім того, існує бібліотека pythonOCC, що забезпечує прив'язку до середовища 3D-моделювання та симуляції OpenCascade.

Для роботи з растровою графікою використовується бібліотека Python Imaging Library.

4.2 Математичні обчислення з використанням бібліотеки NumPy

Бібліотека NumPy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник NumPy, Numeric, був спочатку створений Jim Hugunin. NumPy — відкрите програмне забезпечення і має багато розробників.

Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C.

NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

4.3 Візуалізація результатів за допомогою бібліотеки `matplotlib`

Бібліотека `Matplotlib` на мові програмування `Python` створена для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримувані зображення можуть бути використані як ілюстрації в публікаціях.

`matplotlib` написана і підтримується в основному Джоном Хантером і поширюється на умовах BSD-подібної ліцензії. Зображення, які генеруються в різних форматах, можуть бути використані в інтерактивній графіці, наукових публікаціях, графічному інтерфейсі користувача, веб-додатках, де потрібно будувати діаграми. В документації автор зізнається, що `Matplotlib` починався з імітування графічних команд `MATLAB`, але є незалежним від нього проектом.

Версія 1.2.0 — остання стабільна — потребує `Python` версії від 2.6 і вище і версію `NumPy` від 1.4 і вище.

Бібліотека `Matplotlib` побудована на принципах ООП, але має процедурний інтерфейс `pylab`, який надає аналоги команд `MATLAB`.

`Matplotlib` є гнучким, легко конфігурованим пакетом, який разом з `NumPy`, `SciPy` і `Python` надає можливості, подібні до `MATLAB`. В даний час пакет працює з декількома графічними бібліотеками, включаючи `wxWindows` і `PyGTK`.

Пакет підтримує багато видів графіків і діаграм:

- Графіки (line plot)
- Діаграми розсіювання (scatter plot)
- Стовпчасті діаграми (bar chart) і гістограми (histogram)
- Секторні діаграми (pie chart)
- Діаграми «Стовбур-листя» (stem plot)
- Контурні графіки (contour plot)
- Поля градієнтів (quiver)
- Спектральні діаграми (spectrogram)

Користувач може вказати осі координат, сітку, додати підписи і пояснення, використовувати логарифмічну шкалу або полярні координати.

Нескладні тривимірні графіки можна будувати з допомогою набору інструментів (toolkit) `mplot3d`. Існують і інші набори інструментів: для картографії, для роботи з Excel, утиліти для GTK та інші.

З допомогою `Matplotlib` можна створювати і анімовані зображення.

Набір підтримуваних форматів зображень, векторних і растрових, можна отримати з словника `FigureCanvasBase.filetypes`. Типові підтримувані формати:

- Encapsulated PostScript (EPS)
- Enhanced Metafile (EMF)
- JPEG
- PDF
- PNG
- PostScript
- RGBA («сирий» формат)
- SVG
- SVGZ
- TIFF

Крім того, на основі класів пакету можна створювати й інші модулі. Наприклад, для генерування спарклайнів.

4.4 Розробка підсистеми збереження інформації

У дипломній роботі для розробки підсистеми збереження інформації використовуємо документо-орієнтовану систему керування базами даних з відкритим кодом, яка не потребує опису схеми таблиць.

4.4.1 Збереження інформації до бази даних MongoDB

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3. MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

Основні можливості MongoDB:

- Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних)
- Досить гнучка мова для формування запитів
- Динамічні запити
- Повна підтримка індексів
- Профілювання запитів
- Швидкі оновлення «на місці»
- Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео
- Журналізація операцій, що модифікують дані в БД

- Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг
- Може працювати відповідно до парадигми MapReduce

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS. Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave. Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave. Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB — продукт досить молодий, і відтак у ньому зустрічаються пб1омилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості)[10]. Компанія-розробник надає платні підтримку, хостинг, консультації.

4.4.2 Формат даних JSON

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML(використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript об'єкти.

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси). Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

JSON будується на двох структурах:

- Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.
- Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

- Об'єкт — це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом }. Кожне значення слідує за : і пари назва/значення відділяються комами.
- Масив — це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.
- Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.
- Рядок — це послідовність з нуля або більше символів Unicode, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком.

4.5 Висновки до четвертого розділу

1. Розглянуті основні засоби розробки інформаційної системи, яка за допомогою нового принципу моделювання (застосування методу дискретних особливостей) по зручності користування, розумінню та швидкодії перевищую існуючі аналоги .

5. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ

Програмне забезпечення розроблялось в середовищі Pycharm 2019 з використанням останнього Python 3.7.3.

Система складається з двох проектів:

- проект бібліотеки класів, написано на мові Python;
- проект інтерфейсу користувача, написано на мові Python з використанням бібліотеки matplotlib.

Також широко використовуються JSON-документи для збереження конфігурацій та даних в базу даних MongoDB.

5.1 Функціональна модель

Для вирішення поставленої задачі було побудовано функціональну модель системи прикладних програм для моделювання двовимірної течії при проектуванні крилового профілю в докритичних режимах (рисунок 5.1). Вона складається з трьох частин, які мають окрему будову призначення та функціональність.

Побудову системи функцій введення даних можна побачити (рисунок 5.2). Модель відповідає за побудову геометрії крилового профілю, навколо якого перебігає процес обтікання та розподілу тиску, тобто оцифровку крилового.

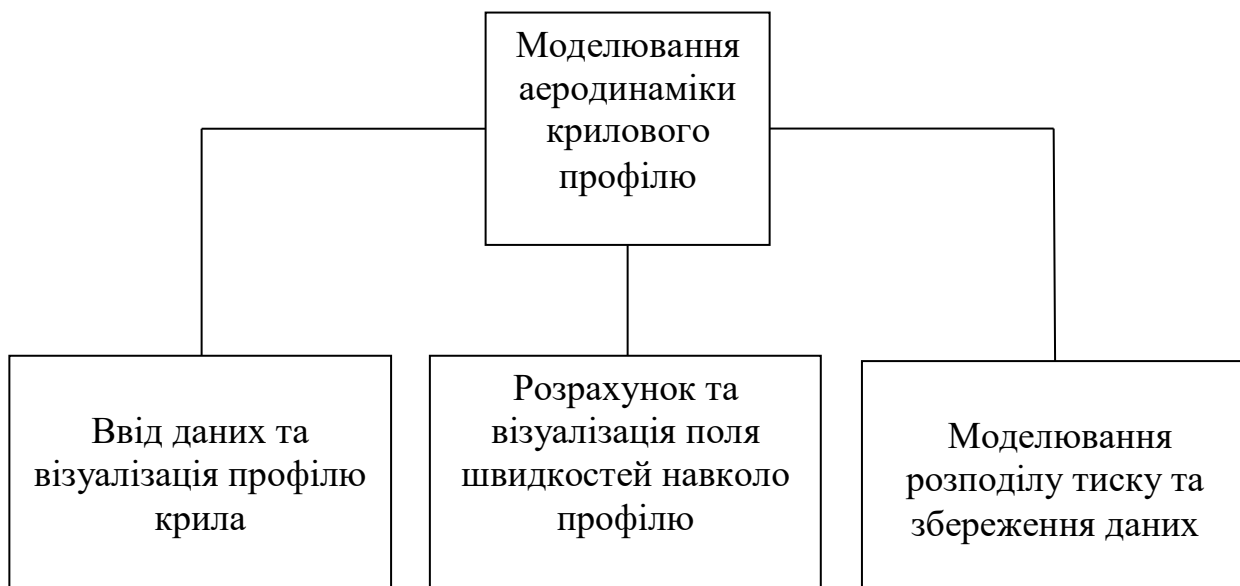


Рисунок 5.1 – Функціональна модель

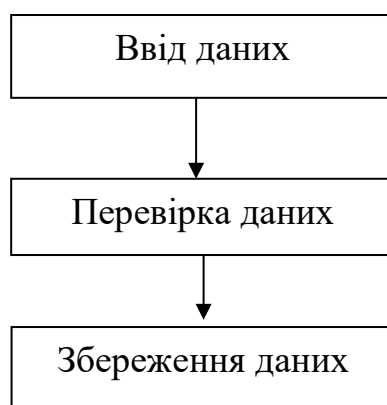


Рисунок 5.2 – Ввід даних

Одна з можливостей у функціях введення даних є перевірка правильності введення. Це надає програмному продукту можливість швидко реагувати на невірно внесені координати.

Функція збереження даних дає можливість запису та збереження, для подальшого використання у розробці та процесумодельованні крилового профілю.

за розрахунок функції току навколо профілю крила, геометрія якої сформована точками колокації та системою нерухомих точкових вихрів, а також візуалізацією траєкторії ліній току на екрані (рисунок 5.3) відповідає система функцій розрахунку та візуалізації поля швидкостей

Вона наділена такими функціями: зчитування даних з файлу; інтегрування рівнянь руху; формування правої частини рівнянь руху; формування СЛАР; розв'язок СЛАР; візуалізація функції току.

Функція зчитування даних з файлу відповідає за діалог між файлом в якому містяться дані про криловий профіль та екранною формою, на якій відображаються координати границі течії, в межах якої розраховується значення функції току, після зчитування інформації з файлу.

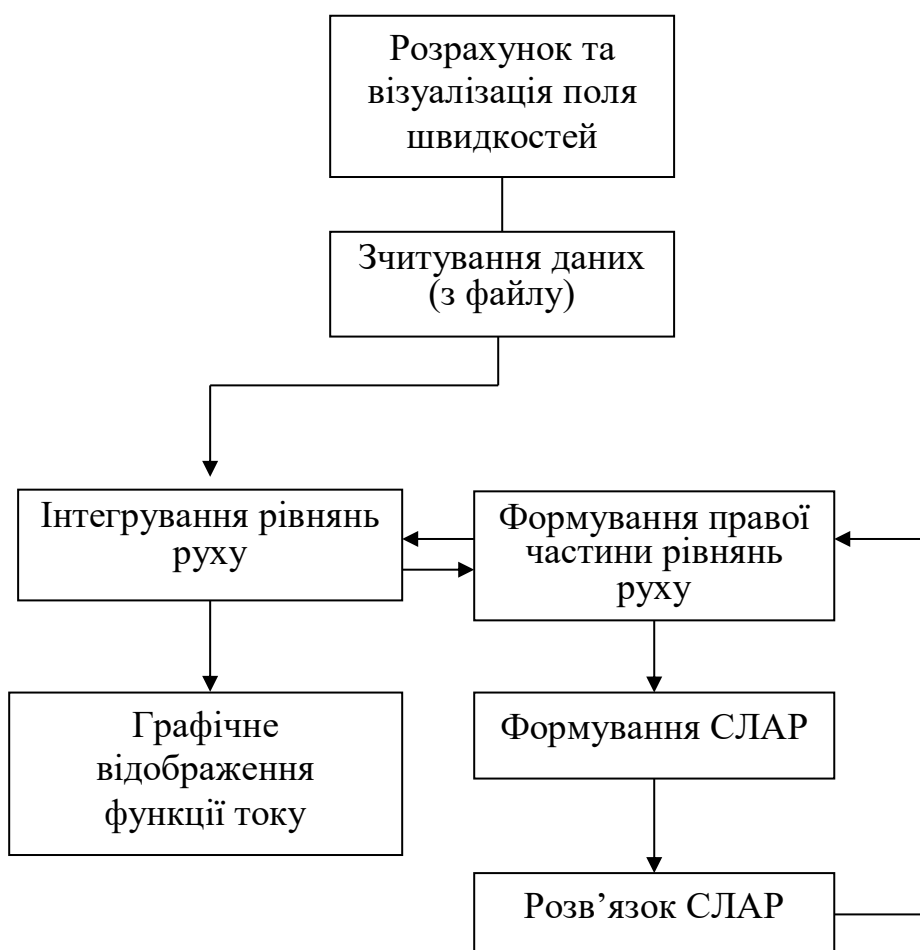


Рисунок 5.3 – Розрахунок та візуалізація поля швидкостей

Функція формування правої частини рівнянь руху генерує праву частину рівняння руху газу так як воно представляє собою задачу Коші з початковими умовами рідких частинок. При цьому в правій частині рівняння міститься вираз, який визначає поле швидкості в тій точці, що знаходиться.

Функція формування СЛАР відповідає за формування системи лінійних алгебраїчних рівнянь відносно невідомих інтенсивностей Γ_i точкових вихрив, застосовуючи умову рівності значень функції току. Розв'язавши її ми отримаємо Γ_i , де $i=1, \dots, N$, які дозволяють визначити розподіл функції току $\Psi(x, y)$.

Функція інтегрування рівнянь руху відповідає за інтегрування рівнянь руху газу. Так як до інтегрування рівняння руху досить високі вимоги щодо точності, оскільки траєкторії руху рідких частинок можуть мати динамічну нестійкість. Для наближення до мінімуму появи обчислювальних помилок в роботі застосовується чисельний метод Рунге-Кутта порядку.

Функція графічного відображення функції току відповідає за графічну візуалізацію траєкторій ліній току по координатам які були отримані після виконання функції інтегрування рівняння руху.

На рисунку 5.4 зображена система функцій розрахунку, візуалізації та збереження , яка відповідає за розподілу тиску на поверхню крила при різних кутах атаки та збереження інформації, яка отримана під час дослідів .

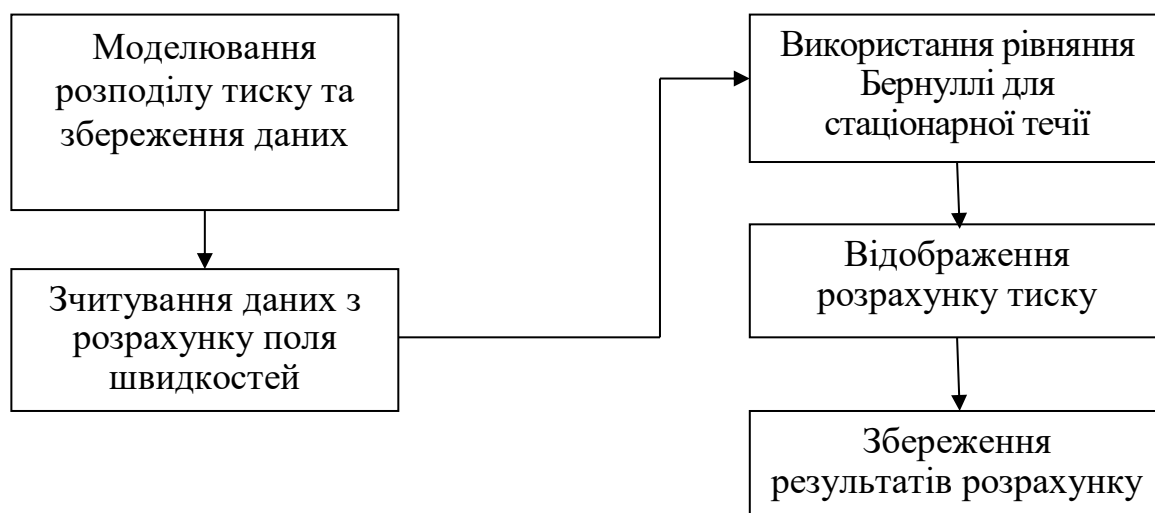


Рисунок 5.4 – Моделювання розподілу тиску та збереження даних

Функція розрахунку тиску на поверхню крила використовує рівняння Бернуллі для стаціонарної течії та попередні розрахунки розподілу поля швидкостей навколо крилового профілю. Наступні функції потрібні для збереження результатів.

Побудова функціональної моделі дала можливість визначити як архітектуру програмного продукту так і визначити основні функціональні проблеми під час створення програмного комплексу спрямованого на моделювання аеродинаміки крилового профілю в до критичних режимах. Далі висвітлюється функціональна модель у повному вигляді (рисунок 5.5).

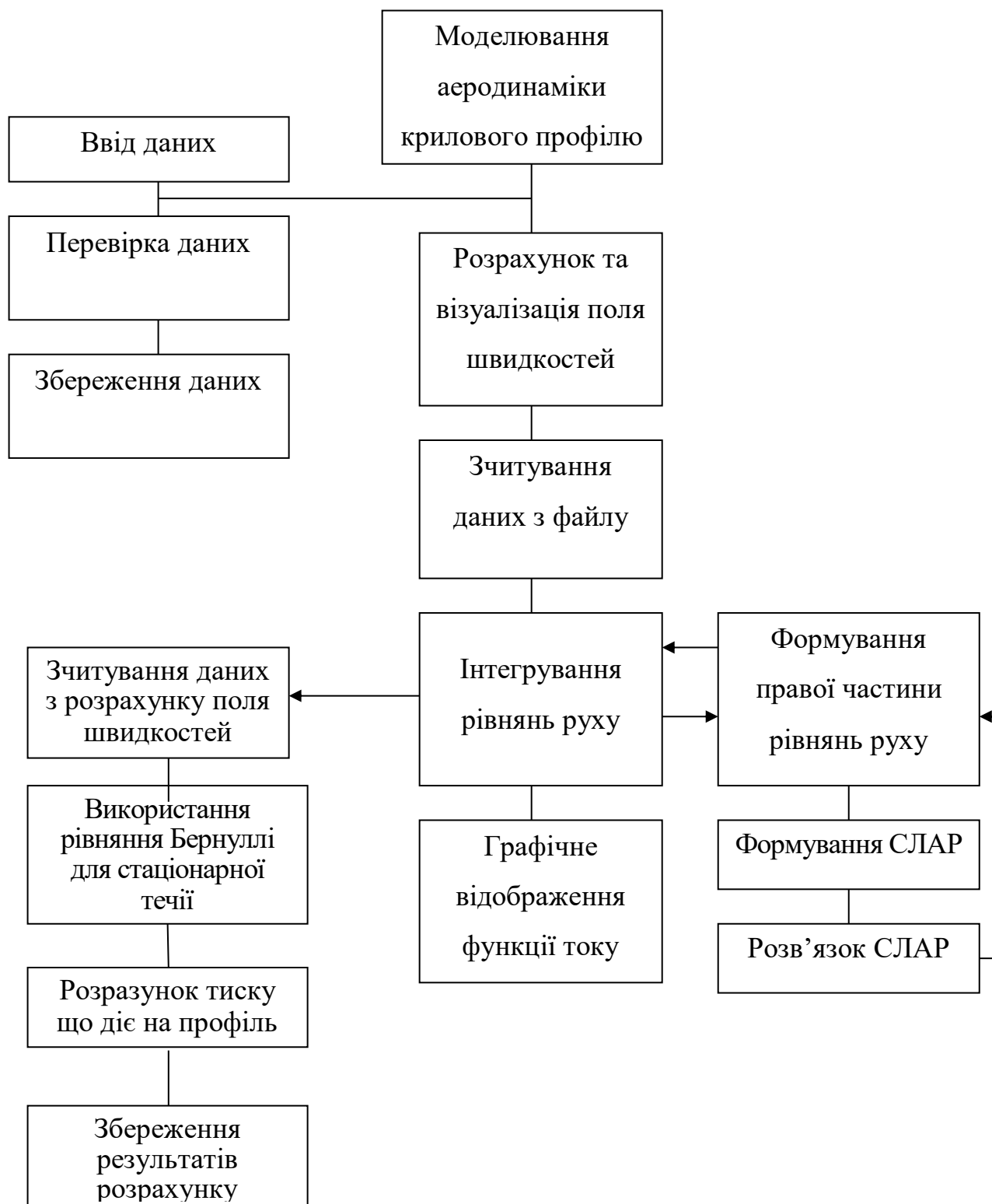


Рисунок 5.5 - Загальний вигляд функціональної моделі

5.2 Внутрішнє представлення даних

Перша підпрограма отримує дані введені користувачем і на їх основі створює базу даних вихорів і маркерів – масив координат, який і буде використовуватись надалі.

База даних вихорів і маркерів має структуру, яка вказана нижче (рисунок 5.6):



Рисунок 5.6- Структура бази даних

де n_v – кількість вихорів, n_m – кількість маркерів.

Друга підпрограма отримує на вхід кількість вихорів, масив координат вихорів і маркерів, значення функцій току у точках колокації на виході отримуємо значення інтенсивностей точкових вихрів, та масив координат – значення функції току в протоці у довільній точці і по ним будується розповсюдження течії навколо профілю крила, тобто її структура (лінії току).

У третій підпрограмі на вхід потрапляють кількість вихорів, масив координат вихорів і маркерів, значення функцій току у точках колокації, геометричні дані крилового профілю в поперечному перерізі. На виході отримуємо масив розподілу тиску в кожній точці навколо профілю крила.

5.3 Структура програми

Перед початком створення програмного забезпечення, необхідно створити діаграму прецедентів (рисунок 5.7). Це дозволяє виділити основних виконавців системи та функції, які програма повинна виконувати.

Нижче представлена структура програмного продукту, яка складається з головного модуля, модулів введення початкових даних – «Візуалізація крилового профілю», розрахунку та візуалізації поля швидкостей – «Попередній аналіз структури течії», моделювання, збереження та розрахунку - «Моделювання процесів адвекції по поверхні течії» (рисунок 5.8).

Головний модуль програмного продукту відповідає за запуск на виконання прикладних задач.

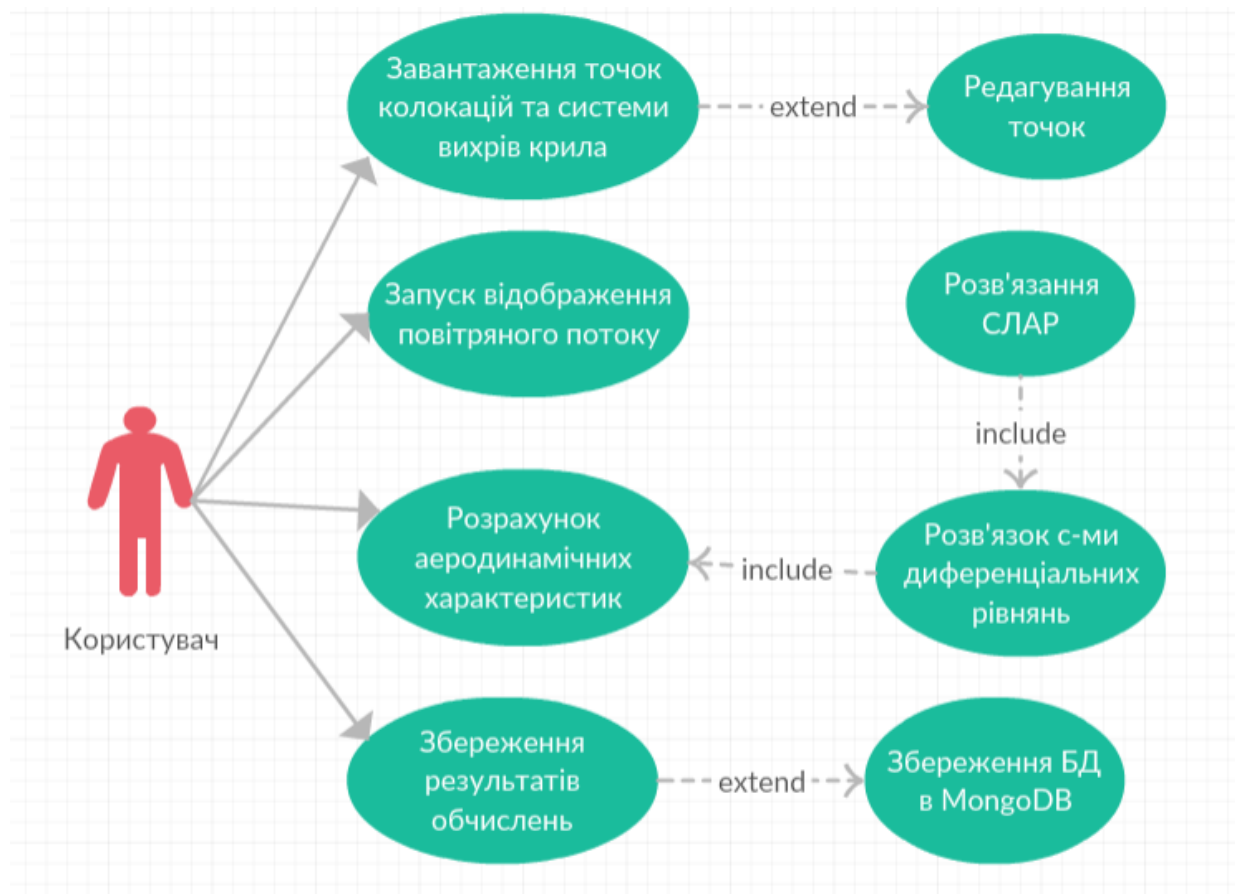


Рисунок 5.7 – UML діаграма прецедентів

Модуль введення даних відповідає за нанесення координат на екранну форму та їх візуалізацію, редагування (приведення до потрібного формату) і перевірку початкових даних задачі.

Наступним на виконання запускається модуль розрахунку та візуалізації поля швидкостей. Він відповідає за розрахунок координат функції току та візуалізацію розповсюдження ліній току навколо крилового профілю. Так як рівняння руху току являє собою задачу Коші, тому для його розв'язання (методом Рунге-Кутта-Фельдберга) в цьому модулі використовується функція RungeKutta. Її параметрами являються: значення функції току у точках колокації, значення інтенсивностей отримані після розв'язку системи лінійних алгебраїчних рівнянь (методом Гауса) та масив координат. За розв'язання СЛАР відповідає функція Gauss. Функція Velocity

відповідає за формування правої частини рівняння руху для задачі Коші. Функція візуалізація в даному модулі відбувається за допомогою власного графічного класу BodyaGraf.cs.



Рисунок 5.8 – Структура програмного продукту

Наступний модуль відповідає за моделювання процесу розподілу тиску на поверхню крилового профілю. В ньому розраховується розподіл тиску в кожній точці навколо профілю крила, використовуючи попередні розрахунки рівняння руху газу і рівняння Бернуллі для стаціонарної течії. За розрахунок тиску відповідає функція Pressure.

На рисунку 5.9 представлена діаграма послідовностей, яка описує взаємодію об'єктів. На діаграмі зображуються лише ті об'єкти, які безпосередньо беруть участь у взаємодії.

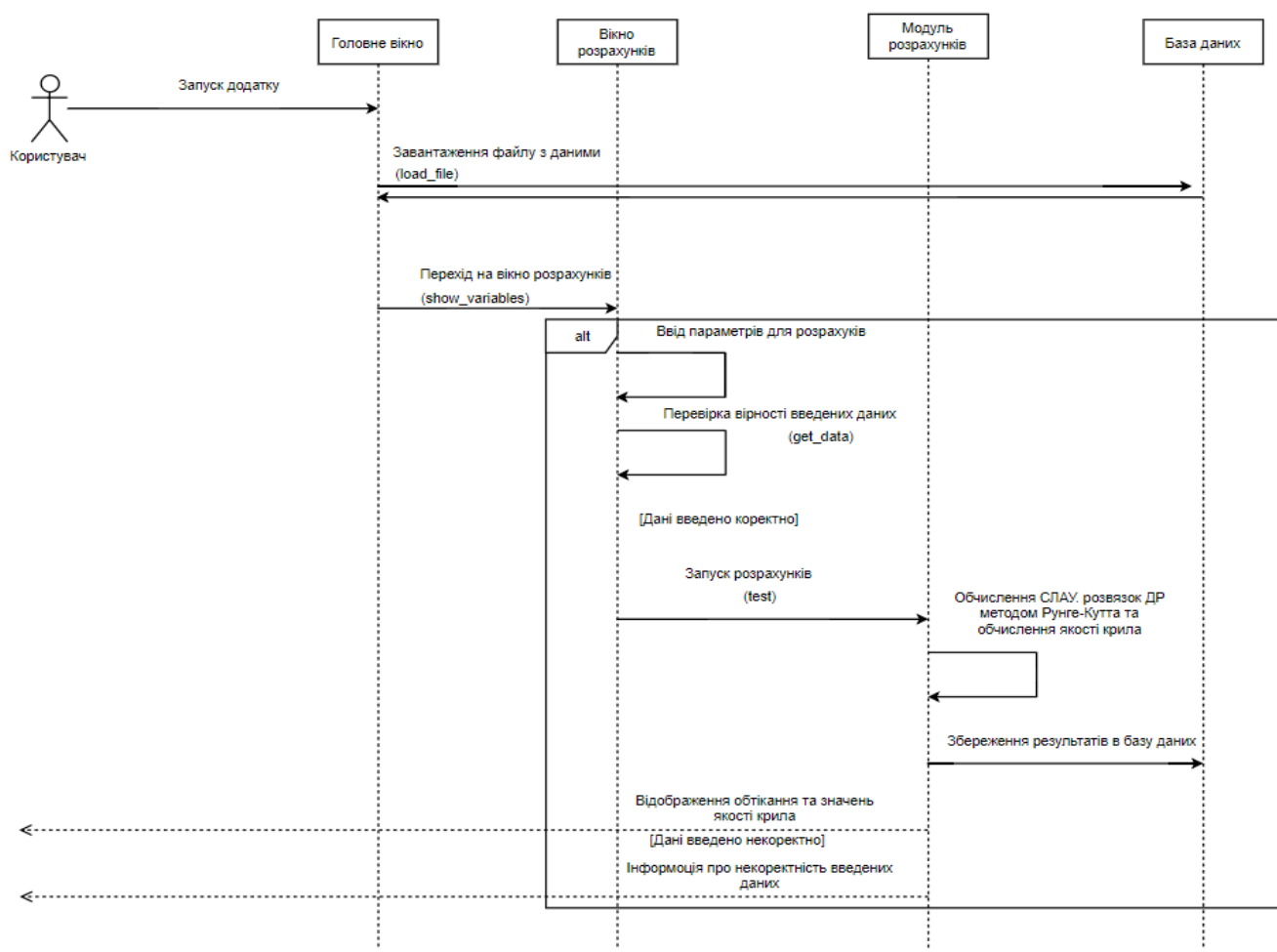


Рисунок 5.9 – діаграма послідовностей програмної системи

5.4 Висновки до п'ятого розділу

1. Було розроблено функціональну модель системи прикладних програм, щоб змодельовати двовимірну течію при проектуванні процесу обтікання крилового профілю на малих швидкостях.

2. Представлена структура програмного продукту, що складається з головного модуля та модулів введення початкових даних – «Візуалізація крилового профілю», розрахунку та візуалізації поля швидкостей – «Попередній аналіз течії», моделювання та збереження розрахунку – «Моделювання процесів адвекції по поверхні течії».

6. ПРИНЦИП РОБОТИ З ПРОГРАМНОЮ СИСТЕМОЮ

6.1 Системні вимоги

Для роботи з програмою, потрібна наявність інстальованого .Python, версії 3.7.3, MongoDB та Robo 3T 1.3.1. Програмна система не потребує встановлення, але для роботи потребує файл з геометричними даними крилового профілю.

Для нормальної роботи програмного комплексу необхідно наступне апаратного забезпечення:

- процесор Intel Pentium з частотою не менше 1.6 GHz;
- оперативна пам'ять в об'ємі 1 Гб.

При запуску програми, що відбувається запуском відповідного виконуючого файлу, який знаходиться у каталозі системи, відображається головне вікно (рисунок 5.1).

6.2. Порядок роботи користувача з системою

Головна форма дозволяє провести наступні дії:

1. завантажити дані з геометрією крила;
2. змодельовати поле швидкостей та розподіл тиску на крило.

Для початку роботи завантажуюмо файл з координатами точок геометрії крила.. Для проведення тестування обрано геометрію крилового профілю NASA – 0012. Після цього потрібно натиснути другу кнопку для виконання задачі моделювання та розподілу тиску на крило (рисунок 5.1).

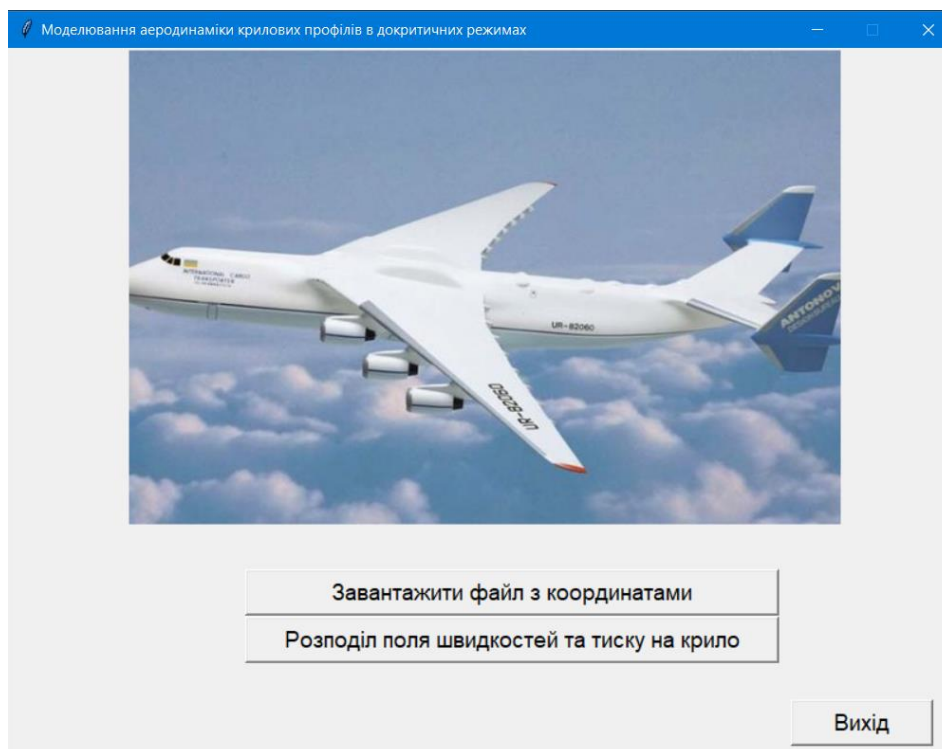


Рисунок 6.1 – Головне вікно програми

Для продовження роботи необхідно заповнити параметри згідно яких буде проводитися візуалізація результату обтікання: U_0 – швидкість потоку повітря (м/с), та кут повороту крила (градуси) відносно початку осі координат, після чого обрати завантажений файл з координатами геометрії профілю (рисунок 6.2).

Рисунок 6.2 – Заповнення параметрів

Після вище перерахованих дій відображається обраний профіль крила під встановленим кутом нахилу, будує обтікання крила (поле швидкостей) для обрахунку якості крила, а в лівій частині вікна бачимо розрахований тиск на крило (Y – вектор напрямку тиску по осі Oy , Q – по осі Ox) та якість крила K (рисунок 6.3)

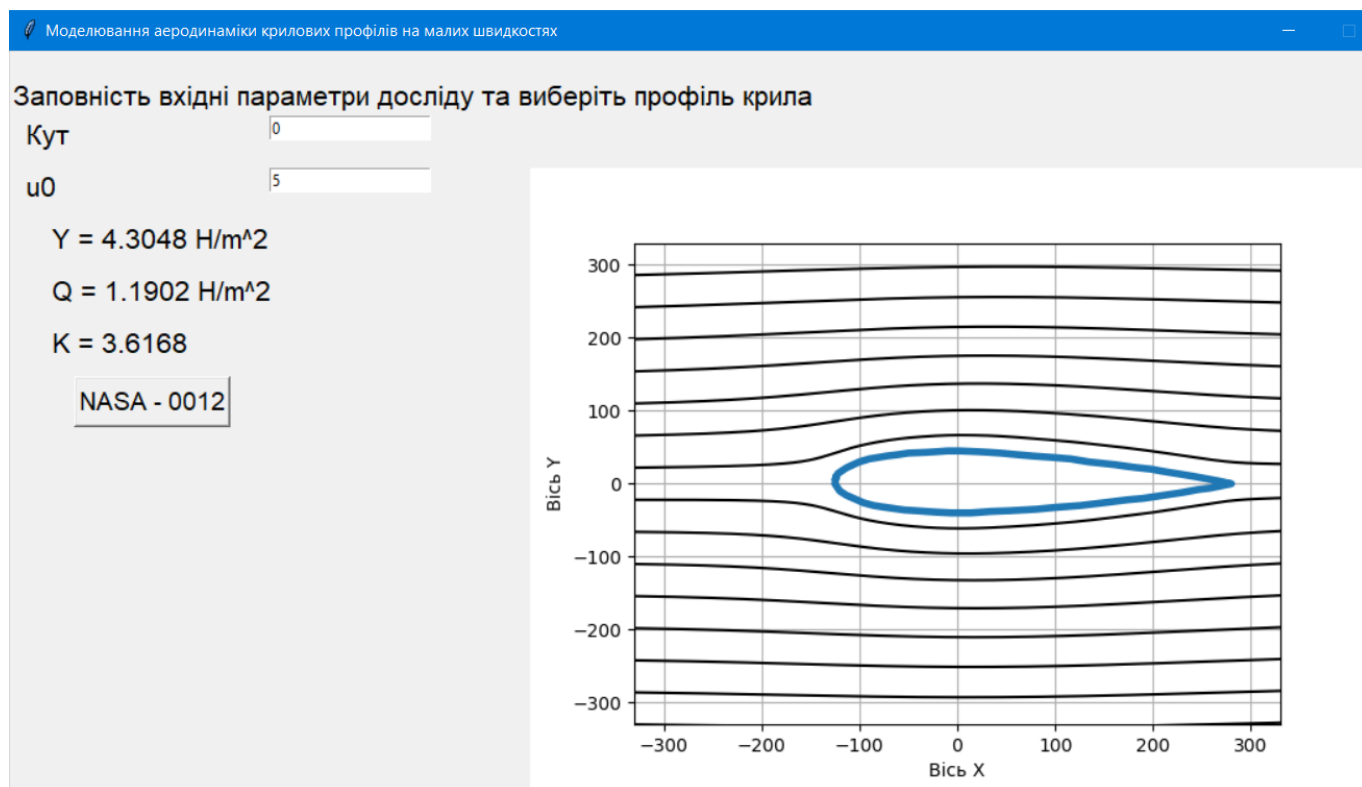


Рисунок 6.3 – Обтікання профілю крила та обрахунок його якості

Користувач має змогу змінювати кут нахилу крила, що продемонстровано на рисунку 6.4.

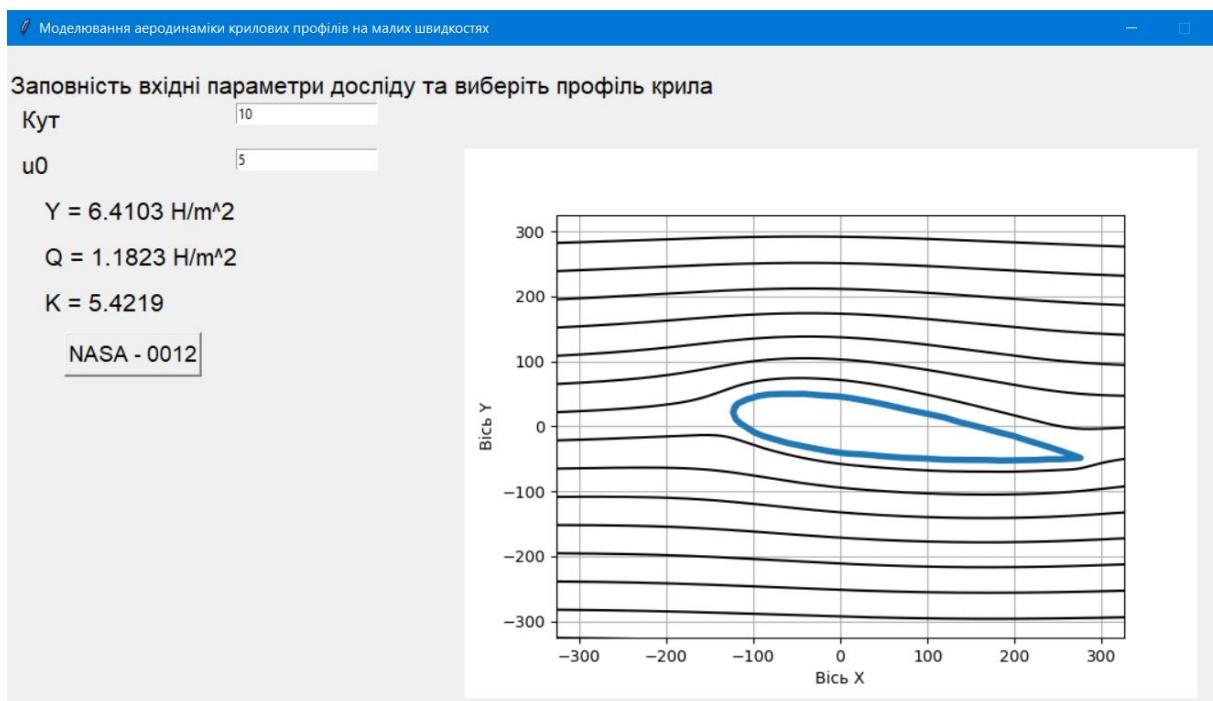


Рисунок 6.4 – Зміна кута нахилу

6.3 Висновки до шостого розділу

1. Визначені системні вимоги для роботи з програмним продуктом.
2. Написано інструкцію для роботи користувача з програмою, щоб змоделювати процес обтікання крилового профілю та визначити його якість.

ВИСНОВКИ

В процесі виконання дипломної роботи було:

1. Досліджено та проаналізовані задачі аеродинаміки
3. Розглянуто задачу процесу моделювання поля швидкостей течією ідеального нестисливого газового потоку та розподілу тиску на криловий профіль.
4. Проведено огляд методів і засобів рішення прикладної задачі аеромеханіки.
5. Проаналізовано існуючі програмні рішення.
6. Та розроблено програмний комплекс, який дозволяє для заданої геометрії поперечного перерізу крила розрахувати аеродинамічні характеристики (якість крила). Модулі системи написані на мові Python в програмному середовищі PyCharm Community Edition з використанням бібліотеки matplotlib
7. Проведена демонстрація розробленого програмного продукту, який дозволяє моделювати процес обтікання заданої геометрії потоком повітря на малих швидкісних режимах, змінюючи кут атаки крила. Дана система спрощує процес моделювання процесу обтікання для визначень аеродинамічних характеристик.

СПИСОК ДЖЕРЕЛ

1. Аерогідрогазодинаміка: підручник / В. Г. Лебедь, Ю. І. Миргород, Є. О. Українець. — Х. : ХУПС ім. Івана Кожедуба, 2011. — 415 с.
2. Аеродинаміка вентиляції: навч. посіб. для студ. вищих навч. закл. / С. С. Жуковський, В. Й. Лабай ; Національний ун-т «Львівська політехніка». — Л. : Видавництво Національного університету «Львівська політехніка», 2003. — 370 с.: рис., табл. — Бібліогр.: с. 365—370.
3. Аеродинаміка літальних апаратів / Харків: ХУПС ім. Івана Кожедуба, 2015.
4. Аеродинаміка літальних апаратів: підруч. для студ. вищ. навч. закл. І—ІІ рівнів акредитації, що навч. за спец. «Виробництво авіаційних та ракетно-космічних апаратів», «Виробництво авіаційних літальних апаратів» / Г. Н. Котельніков [та ін.] ; ред. Ю. М. Терещенко. — К. : Вища освіта, 2002. — 254 с.: рис.
5. Василенко С. М., Кулінченко В. Р., Шевченко О. Ю., Піддубний В. А. Гідрогазодинаміка. — К.: Кондор-Видавництво, 2016. — 676 с.
6. Завойко Б. М., Лещій Н. П. Технічна механіка рідин і газів: основні теоретичні положення та задачі : навч. посібник для студ. інж.-техн. спец./За ред. В. М.
7. Іщенко С. А., Трюхан О. Н. Принципи польоту. — Київ: НАУ, 2011. — 420 с.
8. Колчунов В. І. Теоретична та прикладна гідромеханіка: Навч. Посібник. — К.: НАУ, 2004. — 336 с.
9. Основи аеродинаміки і тепломасообміну: Навч. посібник / Озарків І. М., Сорока Л. Я., Грицюк Ю. І. — К.: ІЗМН, 1997. — 280 с.
10. Основи аеродинаміки і динаміки польоту: навч.. пос. для вищ. навч. закл / В. Е. Касторский ; Рига: РКИИГА; Колос, 2010. — 105 с
11. Abascal, A. J., Castanedo, S., Medina, R., and Liste, M. (2010). Analysis of the reliability of a statistical oil spill response model,
12. Marine Pollution Bulletin, 60 (11), 2099-2110.

13. Andrade, M. M. N., Szlafsztein, C. F., Souza-Filho, et al. (2010). A socioeconomic and natural vulnerability index for oil spills in an
 14. Amazonian harbor: A case study using GIS and remote sensing, *Journal of Environmental Management*, 91 (10), 1972-1980.
15. Anilkumar, P. P., Varghese, K., and Ganesh, L. S. (2010). Formulating a coastal zone health metric for landuse impact management in urban coastal zones, *Journal of Environmental Management*, 91 (11), 2172-2185.
16. Dritschel, D.G. (1989). Contour dynamics and contour surgery: numerical algorithms for extended, high-resolution modelling of vortex dynamics in two-dimensional, inviscid, incompressible flows, *Computer Physics Reports*, 10 (3), 77-146.
17. Gourjii, A.A., Meleshko, V.V. and van Heijst, G.J.F. (2003). Chaotic regimes of fluid stirring by point vortex pair in the circle [in book: “Fundamental and applied problems in vortex theory”, ed. Borisov A.B. et al.], Institute of Computing Researches: Moscow, 441-467 [in Russian].
18. Gourjii, A.A. and Cherniy, D.I. (2009). Adaptive method of discrete singularities in the advection problem of passive impurities by sea flows, *Applied hydromechanics*, 11 (2), 30-39 [in Russian].
19. Gourjii, A.A. (2011). Chaotic fluid advection by vortex flows: DS Thesis in Physics and Mathematics, Institute of Hydromechanics of National Academy of Science: Kyiv [in Russian].
20. Gurzhiy, A.A. and Meleshko, V.V. (1992), Two-dimensional motion that particles of an ideal fluid execute in the velocity field of vortex pair, *Fluid Mechanics – Soviet Research*, 21 (1), 106-112.
21. Lehr, W.J., Jones, R., Evans, M. et al (2002) Revisions of the ADIOS oil spill model, *Environmental Modelling & Software*, 17 (2), 189-197.

23. Liu, X., Writz, K. W., Kannen, A., and Kraft, D. (2009). Willingness to pay among households to prevent coastal resources from polluting by oil spills: A pilot survey. *Marine Pollution Bulletin*, 58 (10), 1514-1521.
24. Liu, X. (2010). Integrated modeling of oil spill response strategies: a coastal management case study. *Environmental Science & Policy*, 13 (5), 415-422.
25. Nittis, K., Perivoliotis, L., Korres, G., et al. (2006). Operational monitoring and forecasting for marine environmental applications in the Aegean Sea, *Environmental Modelling & Software*, 21 (2), 243-257.

ДОДАТОК А

Моделювання процесу обтікання крилового профілю на малих швидкостях

Специфікація

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТР5165_19Б

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР5165_19Б	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР5165_19Б 12-1	main.py select_dots_page.py file_upload.py calculation_engine.py	Основні компоненти
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР5165_19Б 13-1	Додаток В.doc	Опис програмного модуля

ДОДАТОК Б

Моделювання процесу обтікання крилового профілю на малих швидкостях

Текст програми

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_TR5165_19Б 12-1

Аркушів9

Київ 2019

```

from tkinter import *
from select_dots_page import show_variables
from PIL import Image, ImageTk
from file_upload import load_file

from pymongo import MongoClient

def window_deleted():
    root.quit()

client = MongoClient()
client = MongoClient('mongodb://localhost:27017/')
db = client["diploma"]
collection = db['dots']
db.drop_collection('dots')

import numpy
from numpy import*

__all__ = ['CalculationEngine']

class CalculationEngine:

    #Конструктор
    def __init__(self, points_x, points_y, whirlwind_x, whirlwind_y, alpha, u0):
        self.points_x = points_x
        self.whirlwind_x = whirlwind_x
        self.points_y = points_y

```

```

self.whirlwind_y = whirlwind_y
self.alpha = alpha
self.u0 = u0
self.p = 1.225

```

#Функція для переведення градусів в радіани

```
def convert_degree_to_radians(self):
```

```

    rads = radians(self.alpha)
    self.alpha = rads

```

#Функція для повороту крила

```
def prepare_dots(self, list_x, list_y):
```

```

    new_x = []
    new_y = []
    for i in range(len(list_x)):
        new_x.append(list_x[i] * math.cos(self.alpha) + list_y[i] *
math.sin(self.alpha))
        new_y.append(-1 * list_x[i] * math.sin(self.alpha) + list_y[i] *
math.cos(self.alpha))
    return new_x, new_y

```

#Функція для обчислення СЛАР

```
def calculate_slar(self):
```

```

    left_side = numpy.array(self.get_left_side())
    right_side = numpy.array(self.get_right_side())
    self.slar_result = numpy.linalg.solve(left_side, right_side)

```

```
#ln([x - xv] ** 2 + [y - yv] ** 2)
```

```
def get_format_for_slar(self, current_point_x, whirlwind_x, current_point_y,
whirlwind_y):
```

```

func = (current_point_x - whirlwind_x) ** 2 + (current_point_y - whirlwind_y)
** 2

return math.log(func)

#Визначаємо ліву частину для СЛАР
def get_left_side(self):
    left_side = []
    for i in range(len(self.points_x)):
        array = []
        for j in range(len(self.points_x)):
            array.append(self.get_format_for_slar(self.points_x[i],
self.whirlwind_x[j],
self.points_y[i], self.whirlwind_y[j]))

        left_side.append(array)
    return left_side

#Визначаємо праву частину для СЛАР
def get_right_side(self):
    return [-1 * 4 * math.pi + y * self.u0 for y in self.points_y]

#Функція u(x,y)
def get_u(self, x, y):
    u_sum = 0
    for i in range(len(self.whirlwind_x)):
        u_sum += self.slar_result[i] * (self.whirlwind_y[i] - y) / ((self.whirlwind_x[i]
- x) ** 2 + (self.whirlwind_y[i] - y) ** 2)
    return 30 * u_sum / (4 * math.pi)

```

#Функція $v(x,y)$

def get_v(self, x, y):

 v_sum = 0

 for i in range(len(self.whirlwind_x)):

 v_sum += self.slar_result[i] * (self.whirlwind_x[i] - x) / ((self.whirlwind_x[i] - x) ** 2 + (self.whirlwind_y[i] - y) ** 2)

 return -30 * v_sum / (4 * math.pi)

#Диф рівняння dxdt для методу рунге кута

def dxdt(self, x, y, u0):

 return self.get_u(x, y) + u0

#Диф рівняння dydt для методу рунге кута

def dydt(self, x, y, u0):

 return self.get_v(x, y)

#Рунге кутта

def rk4(self, x0, y0, n):

 vx = []

 vy = []

 vx.append(x0)

 vy.append(y0)

 x = x0

 y = y0

 h = 0.01

 for i in range(n):

 kx0 = self.dxdt(x, y, self.u0)

 ky0 = self.dydt(x, y, self.u0)


```
kx1 = h * self.dxdx(x + 2 * kx0 / 9, y, self.u0)
```

```
ky1 = h * self.dydy(x, y + 2 * ky0 / 9, self.u0)
```

```
kx2 = h * self.dxdx(x + kx0 / 12 + kx1 / 4, y, self.u0)
```

```
ky2 = h * self.dydy(x, y + ky0 / 12 + ky1 / 4, self.u0)
```

```
kx3 = h * self.dxdx(x + 69 * kx0 / 128 - 243 * kx1 / 128 + 135 * kx2 / 64, y,  
self.u0)
```

```
ky3 = h * self.dydy(x, y + 69 * ky0 / 128 - 243 * ky1 / 128 + 135 * ky2 / 64,  
self.u0)
```

```
kx4 = h * self.dxdx(x - 17 * kx0 / 12 + 27 * kx1 / 4 - 27 * kx2 / 5 + 16 * kx3 /  
15, y, self.u0)
```

```
ky4 = h * self.dxdx(x, y - 17 * ky0 / 12 + 27 * ky1 / 4 - 27 * ky2 / 5 + 16 * ky3  
/ 15, self.u0)
```

```
x = x + kx0 / 9 + 9 * kx2 / 20 + 16 * kx3 / 45 + kx4 / 12
```

```
y = y + ky0 / 9 + 9 * ky2 / 20 + 16 * ky3 / 45 + ky4 / 12
```

```
vx.append(x)
```

```
vy.append(y)
```

```
return vx, vy
```

#Функції для отримання крайніх точок крила - необхідні для встановлення осей

```
def get_xmin(self):
```

```
    return min(self.points_x)
```

```
def get_xmax(self):
```

```
return max(self.points_x)
```

```
def get_ymin(self):
```

```
    return min(self.points_y)
```

```
def get_ymax(self):
```

```
    return max(self.points_y)
```

```
#Обчислення ліній обтікання за допомогою методу рунге кута
```

```
def test(self):
```

```
    self.calculate_slar()
```

```
    xt = -1 * (max(abs(self.get_xmax()), abs(self.get_ymax()), abs(self.get_xmin()),  
abs(self.get_ymin())) +50)
```

```
    yt = -1 * (max(abs(self.get_xmax()), abs(self.get_ymax()), abs(self.get_xmin()),  
abs(self.get_ymin())) +50)
```

```
    ymax = max(abs(self.get_xmax()), abs(self.get_ymax()), abs(self.get_xmin()),  
abs(self.get_ymin())) +50
```

```
    step = (ymax - yt) / 15
```

```
    result = []
```

```
    for i in range(15):
```

```
        x, y = self.rk4(xt, yt,1500)
```

```
        result.append((x, y))
```

```
        yt += step
```

```
    return result
```

```
#Функція для знаходження найближчих ліній обтікання для крила
```

```
def find_lines(self, result):
```

```
    ymax = self.get_ymax()
```

```
    ymin = self.get_ymin()
```

```
    xmax = self.get_xmax()
```

```

xmin = self.get_xmin()
min_line = result[0]
max_line = result[len(result) - 1]
for i in result:
    if self.check_lines(i[0], i[1]):
        for j in range(len(i[0])):
            if xmax > i[0][j] > xmin:
                if ymax > i[1][j] > min_line[1][j]:
                    min_line = i
                    continue
                elif ymax < i[1][j] < max_line[1][j]:
                    max_line = i
                    continue
return min_line, max_line

```

#Функція для перевірки ліній обтікання яка використовується в функції пошуку вище

```

def check_lines(self, xs, ys):
    xmax = self.get_xmax()
    xmin = self.get_xmin()
    ymax = self.get_ymax()
    ymin = self.get_ymin()
    for i in range(len(xs)):
        if ymax > ys[i] > ymin and xmax > xs[i] > xmin:
            return False
    return True

```

#Функція для поррахунку відстанні між 2-ма сусідніми точками

```

def get_l(self, x0, y0, x1, y1):
    return math.sqrt((x1 - x0) ** 2 + (y1 - y0) ** 2)

```

#Функція для обчислення тиску

```
def get_p(self, x, y, speed, func):
```

```
    return self.p * speed ** 2 / 2 - self.p * func(x, y) ** 2 / 2
```

#Функція для обчислення якості крила

```
def cac1_p(self, result):
```

```
    min_line, max_line = self.find_lines(result)
```

```
    min_sum_y = 0
```

```
    min_sum_x = 0
```

```
    max_sum_y = 0
```

```
    max_sum_x = 0
```

```
    for i in range(len(min_line[0]) - 1):
```

```
        min_sum_y = self.get_p(min_line[0][i], min_line[1][i], self.p, self.get_u) *
```

```
self.get_l(min_line[0][i], min_line[1][i], min_line[0][i + 1], min_line[1][i + 1])
```

```
        min_sum_x = self.get_p(min_line[0][i], min_line[1][i], self.p, self.get_v) *
```

```
self.get_l(min_line[0][i], min_line[1][i], min_line[0][i + 1], min_line[1][i + 1])
```

```
        max_sum_y = self.get_p(max_line[0][i], max_line[1][i], self.p, self.get_u) *
```

```
self.get_l(max_line[0][i], max_line[1][i], max_line[0][i + 1], max_line[1][i + 1])
```

```
        max_sum_x = self.get_p(max_line[0][i], max_line[1][i], self.p, self.get_v) *
```

```
self.get_l(max_line[0][i], max_line[1][i], max_line[0][i + 1], max_line[1][i + 1])
```

```
    fx = (min_sum_x + max_sum_x)
```

```
    fy = (min_sum_y + max_sum_y)
```

```
    return fy / fx, fy, fx
```